JC19 Rec'd PCT/PTO 30 MAY 2001

FORM PTO-1390 US DEPARTMENT OF COMMERCE REV. 5-93PATENT AND TRADEMARK OFFICE **TRANSMITTAL LETTER TO THE UNITED STATES DESIGNATED/ELECTED OFFICE (DO/EO/US) CONCERNING A FILING UNDER 35 U.S.C. 371**		ATTORNEYS DOCKET NUMBER **P01,0140** U.S. APPLICATION NO. (if known, see 37 CFR 1.5) 09/856936
INTERNATIONAL APPLICATION NO. **PCT/DE99/03824**	INTERNATIONAL FILING DATE **01 DECEMBER 1999**	PRIORITY DATE CLAIMED **01 DECEMBER 1998**

TITLE OF INVENTION

SOFT DECISION DECODING OF A SCHEDULED CONVOLUTIONAL CODE

APPLICANT(S) FOR DO/EO/US

THOMAS STURM

Applicant herewith submits to the United States Designated/Elected Office (DO/EO/US) the following items and other information:

1. ☒ This is a **FIRST** submission of items concerning a filing under 35 U.S.C. 371.
2. ☐ This is a **SECOND** or **SUBSEQUENT** submission of items concerning a filing under 35 U.S.C. 371.
3. ☒ This express request to begin national examination procedures (35 U.S.C. 371(f)) at any time rather than delay.
4. ☒ A proper Demand for International Preliminary Examination was made by the 19th month from the earliest claimed priority date.
5. ☒ A copy of International Application as filed (35 U.S.C. 371(c)(2)).
   a. ☒ is transmitted herewith (required only if not transmitted by the International Bureau).
   b. ☐ has been transmitted by the International Bureau.
   c. ☐ is not required, as the application was filed in the United States Receiving Office (RO/US)
6. ☒ A translation of the International Application into English (35 U.S.C. 371(c)(2).
7. ☒ Amendments to the claims of the International Application under PCT Article 19 (35 U.S.C. §371(c)(3))
   a. ☐ are transmitted herewith (required only if not transmitted by the International Bureau).
   b. ☐ have been transmitted by the International Bureau.
   c. ☐ have not been made; however, the time limit for making such amendments has NOT expired.
   d. ☒ have not been made and will not be made.
8. ☐ A translation of the amendments to the claims under PCT Article 19 (35 U.S.C. 371(c)(3)).
9. ☒ An oath or declaration of the inventor(s) (35 U.S.C. 371(c)(4)).
10. ☒ A translation of the annexes to the International Preliminary Examination Report under PCT Article 36 (35 U.S.C. 371(c)(5)).

Items 11. to 16. below concern other document(s) or information included:

11. ☒ An Information Disclosure Statement under 37 C.F.R. 1.97 and 1.98; **(PTO 1449, Prior Art, Search Report, 07 References)**.
12. ☒ An assignment document for recording. A separate cover sheet in compliance with 37 C.F.R. 3.28 and 3.31 is included. **(SEE ATTACHED ENVELOPE)**
13. ☒ Amendment "A" Prior to Action and Appendix "A".
    ☐ A SECOND or SUBSEQUENT preliminary amendment.
14. ☒ A substitute specification and substitute specification mark-up.
15. ☐ A change of address letter attached to the Declaration.
16. ☒ Other items or information:
    a. ☒ Submission of Drawings, 4 sheets of drawings, Figures 1-5
    b. ☒ EXPRESS MAIL #EL 843728138 US dated May 30, 2001

531 Rec'd PCT 30 MAY 2001

U.S. APPLICATION NO. (if known, see 37 CFR 1.5) 09/856936	INTERNATIONAL APPLICATION NO PCT/DE99/03824	ATTORNEY'S DOCKET NUMBER P01,0140

17. ☒ The following fees are submitted:

BASIC NATIONAL FEE (37 C.F.R. 1.492(a)(1)-(5):
Search Report has been prepared by the EPO or JPO $860.00

International preliminary examination fee paid to USPTO (37 C.F.R. 1.482) $690.00

No international preliminary examination fee paid to USPTO (37 C.F R. 1.482) but international search fee paid to USPTO (37 C F R. 1.445(a)(2) $710 00

Neither international preliminary examination fee (37 C F R. 1.482) nor international search fee (37 C.F.R. 1.445(a)(2) paid to USPTO $1000 00

International preliminary examination fee paid to USPTO (37 C.F R 1 482) and all claims satisfied provisions of PCT Article 33(2)-(4) $ 100.00

				CALCULATIONS	PTO USE ONLY
ENTER APPROPRIATE BASIC FEE AMOUNT =				$ 860.00	
Surcharge of $130.00 for furnishing the oath or declaration later than ☐ 20 ☐ 30 months from the earliest claimed priority date (37 C F R. 1.492(e)).				$	
Claims	Number Filed	Number Extra	Rate		
Total Claims	06 - 20 =	0	X $ 18.00	$	
Independent Claims	01 - 3 =	0	X $ 80.00	$	
Multiple Dependent Claims			$270.00 +	$	
TOTAL OF ABOVE CALCULATIONS =				$ 860.00	
Reduction by ½ for filing by small entity, if applicable Verified Small Entity statement must also be filed. (Note 37 C.F.R. 1.9, 1.27, 1.28)				$	
SUBTOTAL =				$ 860.00	
Processing fee of $130.00 for furnishing the English translation later than ☐ 20 ☐ 30 months from the earliest claimed priority date (37 CFR 1.492(f)). +				$	
TOTAL NATIONAL FEE =				$ 860.00	
Fee for recording the enclosed assignment (37 C F R. 1.21(h). The assignment must be accompanied by an appropriate cover sheet (37 C F R 3.28, 3.31). $40 00 per property +					
TOTAL FEES ENCLOSED =				$ 860.00	
				Amount to be refunded	$
				charged	$

a. ☒ A check in the amount of $ 860.00 to cover the above fees is enclosed.

b. ☐ Please charge my Deposit Account No. __________ in the amount of $ _______ to cover the above fees. A duplicate copy of this sheet is enclosed.

c. ☒ The Commissioner is hereby authorized to charge any additional fees which may be required, or credit any overpayment to Deposit Account No. **50-1519**. **A duplicate copy of this sheet is enclosed.**

NOTE: Where an appropriate time limit under 37 C.F.R. 1.494 or 1.495 has not been met, a petition to revive (37 C.F.R. 1.137(a) or (b)) must be filed and granted to restore the application to pending status.

SEND ALL CORRESPONDENCE TO:

SCHIFF HARDIN & WAITE
PATENT DEPARTMENT
6600 Sears Tower
233 South Wacker Drive
Chicago, Illinois 60606-6473

CUSTOMER NUMBER 26574

Mark Bergner
SIGNATURE **MARK BERGNER** **(Reg. No. 45,877)**

Dated: May 30, 2001

09/856936

531 Rec'd PCT 30 MAY 2001

BOX PCT
IN THE UNITED STATES DESIGNATED/ELECTED OFFICE
OF THE UNITED STATES PATENT AND TRADEMARK OFFICE
UNDER THE PATENT COOPERATION TREATY--CHAPTER II

PRELIMINARY AMENDMENT A
PRIOR TO ACTION

APPLICANT(S):	Thomas STURM
ATTORNEY DOCKET NO.:	P01,0140
INTERNATIONAL APPLICATION NO:	PCT/DE99/03824
INTERNATIONAL FILING DATE:	01 December 1999
INVENTION:	SOFT DECISION DECODING OF A SCHEDULED CONVOLUTIONAL CODE

Assistant Commissioner for Patents,
Washington D.C. 20231

Sir:

Applicants herewith amend the above-referenced PCT application, and request entry of the Amendment prior to examination on the United States Examination Phase.

IN THE CLAIMS:

On amended page 25:

replace line 1 with --WHAT IS CLAIMED IS:--;

Please replace original claims 1-6 with the following rewritten claims 1-6, referring to the mark-ups in Appendix A.

1. (Amended) A method for decoding a predetermined code word, wherein said code word comprises a number of positions having different values, comprising the steps of:

providing a processor comprising a central processing unit, memory, an input/output interface, and a data bus connecting said central processing unit to said memory and said input/output interface, said processor decoding a predetermined code word;

determining a calculation rule for a soft-output value for each position of said code word, each position of said code word being correlated with said soft-output value, according to the formula

$$L(U_i|y) = \ln\left(\frac{\sum_{c\in\Gamma^i(+1)} \exp\left(-\frac{(y-c)^\top(y-c)}{2\sigma^2}\right)}{\sum_{c\in\Gamma^i(-1)} \exp\left(-\frac{(y-c)^\top(y-c)}{2\sigma^2}\right)}\right), \quad \text{for} \quad i = 1,\ldots,K,$$

where

$L(U_i|y)$ is a safety measure (soft output) for the i-th position of the code word to be determined;

y is a demodulation result to be decoded;

c is a code word;

$\Gamma^i(\pm 1)$ are all code words for $u_i = \pm 1$; and

σ^2 is a variance (channel disturbance);

utilizing a characteristic of a convolutional code, in decoding of said code word, for determining said correlation of said individual positions of said code word from which steps follow of determining states in accordance with a shift register operation, and obtaining a trellis representation from these states;

calculating weights $\mu_q(s)$, for an arbitrary choice of $y \in \mathrm{R}^N$, for nodes (s, q) of said trellis representation by evaluating

$$\mu_q : S \to \mathbb{R},$$

$$s \mapsto \exp\left(-\frac{1}{2\sigma^2}\sum_{j=1}^{n}\left(y_{n(q-1)+j} - C_j(s)\right)^2\right)$$

for $q \in \{1,\ldots,Q\}$.

determining mappings A_m by way of said trellis representation, running through said trellis representation in the natural direction, and calculating the term A_m by

$$A_m(s) = \mu_m(s) \sum_{t\in W(s,V_m)} A_{m-1}(t), \quad \text{for} \quad m \in \mathbb{N}$$

and a starting value

$$A_0(s) = \begin{cases} 1 & : \quad \text{for} \quad s = s_0, \\ 0 & : \quad \text{else} \end{cases}$$

determining mappings B_m by way of said trellis representation, said trellis representation being run through in opposition to a predetermined direction, and calculating the term B_m by

$$B_m(s) = \mu_{Q-m+1}(s) \sum_{t \in T(s, V_{Q-m+2})} B_{m-1}(t), \qquad \text{for} \quad 1 \leq m \leq Q,$$

where

$$B_0(s) = \begin{cases} 1 & : \quad \text{for} \quad s = s_0, \\ 0 & : \quad \text{else} \end{cases}$$

is determined for terminating the recursion; and

determining terms A_α^i by again running through said trellis representation taking into consideration said terms A_m and B_m already determined, according to a relation

$$A_\alpha^i(y) = \sum_{s \in S} A_{j-1}(s) \sum_{t \in T(s, V_j^i(\alpha))} B_{Q-j+1}(t),$$

where $j = \left\lfloor \frac{i-1}{b} \right\rfloor + 1$;

determining K positions of said code word according to

$$L(U_i|y) = \ln\left(\frac{A_{+1}^i(y)}{A_{-1}^i(y)}\right), \quad i = 1, \ldots, K.$$

2. (Amended) The method as claimed in claim 1, wherein said convolutional code has binary state transitions, said method further comprising the steps of:

determining mappings A_m recursively by the equation

$$A_m(s) = \mu_m(s)\left(A_{m-1}\left(\hat{T}(+1,s)\right) + A_{m-1}\left(\hat{T}(-1,s)\right)\right), \quad \text{for} \quad m \in \mathbb{N};$$

determining mappings B_m recursively by the equation

$$B_m(s) = \mu_{Q-m+1}(s)\left(B_{m-1}\left(T(s,+1)\right) + B_{m-1}\left(T(s,-1)\right)\right),$$

for $1 \leq m \leq Q$; and

determining terms A_α^i, $i \in \{1,\ldots,K\}$, $\alpha \in \{\pm 1\}$ according to the equation

$$A_\alpha^i(y) = \sum_{s \in S} A_{i-1}(s) B_{Q-i+1}(T(s,\alpha)).$$

3. (Amended) The method as claimed in claim 1, further comprising the step of:

providing a mobile radio network in which said decoding of a predetermined code word operates.

4. (Amended) The method as claimed in claim 3, wherein said mobile radio network is a GSM network.

5. (Amended) The method as claimed in claim 1, wherein said predetermined code word is a concatenated code word, said method further comprising the steps of:

providing said calculated soft-output values as input data of another decoder.

6. (Amended) An arrangement for decoding a predetermined code word, comprising:

a processor having a central processing unit, memory, an input/output interface, and a data bus connecting said central processing unit to said memory and said input/output interface, said processor being configured for carrying out the method according to claim 1.

REMARKS

The present Amendment revises the specification and claims to conform to United States patent practice, before examination of the present PCT application in the United States National Examination Phase. Pursuant to 37 CFR 1.125 (b), applicants have concurrently submitted a substitute specification, excluding the claims, and provided a marked-up copy. All of the changes are editorial and applicant believes no new matter is added thereby. The amendment, addition, and/or cancellation of claims is not intended to be a surrender of any of the subject matter of those claims.

Early examination on the merits is respectfully requested.

Submitted by,

Mark Bergner (Reg. No. 45,877)

Mark Bergner
Schiff Hardin & Waite
Patent Department
6600 Sears Tower
233 South Wacker Drive
Chicago, Illinois 60606-6473
(312) 258-5779
Attorneys for Applicant

CUSTOMER NUMBER 26574

09/856936
531 Rec'd PCT 30 MAY 2001

Appendix A

Mark Ups for Claim Amendments

This redlined draft, generated by CompareRite (TM) - The Instant Redliner, shows the differences between -

original document : Q:\DOCUMENTS\YEAR 2001\P010140-ITURM-SOFT DECISION CODING\ORIGINAL CLAIMS.DOC

and revised document: Q:\DOCUMENTS\YEAR 2001\P010140-ITURM-SOFT DECISION CODING\AMENDED CLAIMS.DOC

CompareRite found 58 change(s) in the text

Deletions appear as Overstrike text surrounded by []

Additions appear as Bold-Underline text

1. **(Amended)** A method for decoding a predetermined code word, **wherein said**[~~(a) in which the~~] code word comprises a number of positions having different values**, comprising the steps of:**[~~;~~]

[~~(b) in which each position of the~~]**providing a processor comprising a central processing unit, memory, an input/output interface, and a data bus connecting said central processing unit to said memory and said input/output interface, said processor decoding a predetermined** code word [~~is correlated with~~]**;**

determining a calculation rule for a soft-output value[~~, in which the calculation rule for the soft-output value~~] for each position of [~~the~~] **said** code word [~~is determined by~~]**, each position of said code word being correlated with said soft-output value, according to the formula**

$$L(U_i|y) = \ln \left(\frac{\sum\limits_{c \in \Gamma^i(+1)} \exp\left(-\frac{(y-c)^\top (y-c)}{2\sigma^2}\right)}{\sum\limits_{c \in \Gamma^i(-1)} \exp\left(-\frac{(y-c)^\top (y-c)}{2\sigma^2}\right)} \right), \quad \text{for} \quad i = 1, \ldots, K ,$$

where

$L(U_i|y)$ is a safety measure (soft output) for the i-th position of the code word to be determined;

y is a demodulation result to be decoded;

c is a code word;

$\Gamma^i(\pm 1)$ are all code words for $u_i = \pm 1$; **and**

σ^2 is a variance (channel disturbance);

[~~(c) in which the decoding of the code word is determined by the correlation of the individual positions of the code word by~~] utilizing a characteristic of a convolutional code**, in decoding of said code word, for determining said correlation of said individual positions of said code word** from which [~~it follows that~~] **steps follow of determining** states [~~are determined~~] in accordance with a shift register operation, [~~from which states~~] **and obtaining** a trellis representation [~~is obtained;~~] **from these states;**

~~[(d) in which]~~ **calculating weights $\mu_q(s)$**, for an arbitrary choice of $y \in \mathrm{R}^N$, ~~[weights $\mu q(s)$ are calculated for the]~~ **for** nodes (s, q) of ~~[the]~~ **said** trellis representation by evaluating

$$\mu_q : S \to \mathbb{R},$$

$$s \mapsto \exp\left(-\frac{1}{2\sigma^2}\sum_{j=1}^{n}\left(y_{n(q-1)+j} - C_j(s)\right)^2\right)$$

for $q \in \{1,\ldots,Q\}$.

~~[(e) in which]~~ **determining** mappings A_m ~~[are determined by means of the]~~ **by way of said** trellis representation, running through ~~[the]~~ **said** trellis representation in the natural direction, **and calculating** the term A_m ~~[being determined]~~ by

$$A_m(s) = \mu_m(s) \sum_{t \in W(s,V_m)} A_{m-1}(t), \quad \text{for} \quad m \in \mathbb{N}$$

and a starting value

$$A_0(s) = \begin{cases} 1 & : \text{ for } s = s_0, \\ 0 & : \text{ else} \end{cases}$$

~~[(f) in which]~~ **determining** mappings B_m ~~[are determined by means of the]~~ **by way of said** trellis representation, ~~[the]~~ **said** trellis representation being run through in opposition to ~~[the]~~ **a** predetermined direction, **and calculating** the term B_m ~~[being determined]~~ by

$$B_m(s) = \mu_{Q-m+1}(s) \sum_{t \in T(s,V_{Q-m+2})} B_{m-1}(t), \quad \text{for} \quad 1 \le m \le Q,$$

where

$$B_0(s) = \begin{cases} 1 & : \quad \text{for } s = s_0, \\ 0 & : \quad \text{else} \end{cases}$$

is determined for terminating the recursion; **and**

determining terms A^i_α [redacted] ~~[(g) in which terms are determined]~~ by again running through ~~[the]~~ **said** trellis representation taking into consideration ~~[the]~~ **said** terms A_m and B_m already determined, ~~[in accordance with the]~~ **according to a** relation

$$A^i_\alpha(y) = \sum_{s \in S} A_{j-1}(s) \sum_{t \in T(s, V_j^i(\alpha))} B_{Q-j+1}(t),$$

where $j = \left\lfloor \frac{i-1}{b} \right\rfloor$ [redacted] $+1$;

~~[(h) in which the]~~ **determining** K positions of ~~[the]~~ **said** code word ~~[are determined in accordance with]~~ **according to**

$$L(U_i|y) = \ln\left(\frac{A^i_{+1}(y)}{A^i_{-1}(y)}\right), \quad i = 1, \ldots, K.$$

~~[2.]~~**2. (Amended)** The method as claimed in **claim 1, wherein said** ~~[one of the preceding claims,~~

~~(a) in which the]~~ convolutional code has binary state transitions**, said method further comprising the steps of:**~~[;]~~

~~[(b) in which the]~~**determining** mappings A_m ~~[are determined]~~ recursively by **the equation**

$$A_m(s) = \mu_m(s)\left(A_{m-1}\left(\hat{T}(+1,s)\right) + A_{m-1}\left(\hat{T}(-1,s)\right)\right), \quad \text{for} \quad m \in \mathbb{N};$$

determining[

~~(c) in which the~~] mappings B_m [~~are determined~~] recursively by **the equation**

$$B_m(s) = \mu_{Q-m+1}(s)\left(B_{m-1}\left(T(s,+1)\right) + B_{m-1}\left(T(s,-1)\right)\right),$$

for $1 \leq m \leq Q$; **and**

determining[~~(d) in which the~~] terms A^i_α [redacted], $i \in \{1,\ldots,K\}$, $\alpha \in \{\pm 1\}$ [~~are determined in accordance with~~] **according to the equation**

$$A^i_\alpha(y) = \sum_{s \in S} A_{i-1}(s) B_{Q-i+1}(T(s,\alpha)).$$

[~~3.~~]**3. (Amended)** The method as claimed in [~~one of the preceding claims, for use in~~] **claim 1, further comprising the step of:**

providing a mobile radio network **in which said decoding of a predetermined code word operates.**

4. (Amended)[~~.~~

~~4.~~] The method as claimed in claim 3, **wherein said** [~~in which the~~] mobile radio network is a GSM network.

5. **(Amended)** The method as claimed in [~~one of the preceding claims, in which the method is a part of the decoding of~~] **claim 1, wherein said predetermined code word is** a concatenated code [~~in which the~~]**word, said method further comprising the steps of:**

providing said calculated soft-output values ~~[are used]~~ as input data of another decoder.

6. **(Amended)** An arrangement for decoding a predetermined code word, ~~[in which]~~ **comprising:**

a processor ~~[unit is provided which is set up in such a manner that a]~~ **having a central processing unit, memory, an input/output interface, and a data bus connecting said central processing unit to said memory and said input/output interface, said processor being configured for carrying out the** method according to ~~[one of the preceding claims can be carried out by this unit.]~~ **claim 1.**

4|PRTS

09/856936

531 Rec'd PCT/ 30 MAY 2001

SPECIFICATION

TITLE

METHOD AND ARRANGEMENT FOR DECODING A PREDETERMINED CODE WORD

BACKGROUND OF THE INVENTION

FIELD OF THE INVENTION

[0001] The invention relates to a method and an arrangement for decoding a predetermined code word.

DESCRIPTION OF THE RELATED ART

[0002] In the decoding of a code word which has a predetermined number of positions, the information-carrying positions are restored as completely as possible.

[0003] The decoding takes place at the end of the receiver which has received the code word via a disturbed channel. Signals are transmitted, in particular as Boolean values, preferably subdivided into +1 and -1, via the channel where they are subject to a disturbance, and are converted into analog values which can deviate to a greater or lesser extent from the predetermined Boolean values (±1) by a demodulator.

[0004] The general assumption is K positions of binary information ("information bits") without redundancy $u \in \{\pm 1\}^K$, which is mapped into a code word $c \in \{\pm 1\}^N$ by means of systematic block codes or unsystematic block codes by a channel coder. In this arrangement, the code word contains $N - K$ bits (also "check bits") which can be used as redundant information to the N information bits for restoring the information after transmission via the disturbed channel.

[0005] The systematic block code adds to the N information bits $N - K$ check bits which are calculated from the information bits, the information bits themselves remaining unchanged whereas, in the unsystematic block code, the information bits themselves are changed, for example the information is in an operation performed from one to the next position. Here, too, check bits are provided for reconstructing the information hidden in the operations. In the text which follows, in particular, a technically significant variant of unsystematic block codes, the so-called terminated convolutional codes, is considered.

[0006] "Hard" decoding of a correlation of the received code word (with the positions occupied by analog values) i.e., correlating each position with the nearest Boolean value in each case, is a decisive disadvantage since valuable information is lost in this process.

SUMMARY OF THE INVENTION

[0007] It is the object of the invention to determine a decoding of a predetermined code word, with decoding supplying analog values (so-called "soft outputs") which, in particular, can be taken into consideration in the subsequent decoding method and thus provide for high error correction in the transmission of code words via a disturbed channel.

[0008] This object is achieved in accordance with the various embodiments of the method and apparatus discussed below.

[0009] To achieve this object, a method for decoding a predetermined code word is specified in which the code word comprises a number of positions having different values. In this arrangement, encoding has taken place, in particular, by way of a terminated convolutional code. Each position of the code word is correlated with a safety measure (soft output) for a most probable Boolean value by performing the correlation on the basis of a trellis representation. The decoding of the code word is determined by the correlation of the individual positions of the code word.

[0010] A decisive advantage here is that due to the correlation based on the trellis representation, a distinct reduction in complexity compared with a general representation takes place with the result that decoding of the code word (generation of the soft outputs at the positions of the code word) also becomes possible in real time.

[0011] A further development consists in that the decoding rule for each position of the code word is determined by

$$L(U_i|y) = \ln\left(\frac{\sum\limits_{c\in\Gamma^i(+1)} \exp\left(-\frac{(y-c)^\top(y-c)}{2\sigma^2}\right)}{\sum\limits_{c\in\Gamma^i(-1)} \exp\left(-\frac{(y-c)^\top(y-c)}{2\sigma^2}\right)}\right), \quad \text{for } i = 1,\ldots,K, \qquad (1)$$

[0012]

[0013] where

[0014] $L(U_i|y)$ is a safety measure (soft output) for the i-th position of the code word to be determined;

[0015] y is a demodulation result to be decoded;

[0016] c is a code word;

[0017] $\Gamma^i(\pm 1)$ are all code words for $u_i = \pm 1$; and

[0018] σ^2 is a variance (channel disturbance).

[0019] Another further development provides that the equation (1) is solved by utilizing a characteristic of a convolutional code used in the coding (and correspondingly in the decoding) which determines states in accordance with a shift register operation used during the convolution, from which states, in turn, the trellis representation is obtained.

[0020] In an additional further development, the trellis representation is run through in a predetermined direction in order to recursively calculate terms A_m and A_m respectively. Into this calculation rule, node weights $\mu_m(s)$ which are determined by the demodulation result y enter at the nodes (s, m) of the trellis representation. The terms A_m and A_m are described by

$$\tilde{A}_m(E) = \sum_{s \in E} A_m(s), \quad \text{for } m \in \mathbb{N} \tag{2}$$

[0021]

[0022] with

$$A_m(s) = \mu_m(s) \sum_{t \in W(s, V_m)} A_{m-1}(t), \quad \text{for } m \in \mathbb{N} \tag{3}$$

[0023]

[0024] and a starting value

$$A_0(s) = \begin{cases} 1 & : \text{ for } s = s_0, \\ 0 & : \text{ else} \end{cases} \tag{4}$$

[0025]

[0026] A more detailed discussion of the forms of description listed here can also be found in the description of the exemplary embodiment.

[0027] One embodiment provides that mappings B_m are determined by way of the trellis representation, the trellis representation being processed in opposition to the predetermined direction. The term B_m is determined by

$$B_m(s) = \mu_{Q-m+1}(s) \sum_{t \in T(s, V_{Q-m+2})} B_{m-1}(t), \quad \text{for } 1 \leq m \leq Q, \tag{5}$$

[0028]

[0029] where

$$B_0(s) = \begin{cases} 1 & : \text{ for } s = s_0, \\ 0 & : \text{ else} \end{cases} \tag{6}$$

[0030]

[0031] is determined for terminating the recursion.

[0032] Furthermore, terms A^i_α can be determined by again running through the trellis representation taking into consideration the terms A_m and B_m already determined. In particular, the terms A^i_α are determined in accordance with

$$A^i_\alpha(y) = \sum_{s \in S} A_{j-1}(s) \sum_{t \in T(s, V^i_j(\alpha))} B_{Q-j+1}(t). \tag{7}$$

[0033]

[0034] In a further embodiment, the K positions of the decoded code word are determined in accordance with

$$L(U_i|y) = \ln\left(\frac{A^i_{+1}(y)}{A^i_{-1}(y)}\right), \quad i = 1, \ldots, K. \tag{8}$$

[0035]

[0036] In particular, an AWGN (Additive Gaussian White Noise) channel model is used for the derivation. The method presented can also be used for other channel models, especially for channel models used in mobile radio.

[0037] Another embodiment relates to the use of the method in a mobile radio network, especially the GSM network.

[0038] It is also a further development that, after the soft outputs have been determined, there is a "hard" correlation of the analogue values with the Boolean values ± 1. In this arrangement, the nearest Boolean value is in each case determined for correlating the analogue value.

[0039] The soft output values determined can be used as input values for further decoding when concatenated codes are used.

[0040] To achieve the object, an arrangement for decoding a predetermined code word is also specified in which a processor unit is provided which is set up in such a manner that

[0041] 1. the code word comprises a number of positions having different values;

[0042] 2. each position of the code word can be correlated with a soft output value by performing the correlation on the basis of a trellis representation; and

[0043] 3. the decoding of the code word can be determined by the correlation of the individual positions of the code word.

[0044] This arrangement is particularly suitable for performing the method according to the invention or one of its further developments explained above.

BRIEF DESCRIPTION OF THE DRAWINGS

[0045] In the text which follows, exemplary embodiments of the invention will be shown and explained with reference to the drawings, in which:

[0046] Figure 1 is a block diagram showing a representation of digital information transmission;

[0047] Figure 2 is an algorithm in pseudocode notation for progressing in the trellis diagram observing all states for the calculation of node weights;

[0048] Figure 3 is an algorithm in pseudocode notation for determining soft outputs (general case);

[0049] Figure 4 is an algorithm in pseudocode notation for determining soft outputs (special case: binary state transition); and

[0050] Figure 5 is a block diagram of a processor unit.

DETAILED DESCRIPTION OF THE INVENTION

[0051] The text which follows describes in greater detail, first the convolutional code, then the reduction in complexity in the calculation of soft outputs and, finally, an algorithmic translation of the reduction in complexity.

TERMINATED CONVOLUTIONAL CODE

[0052] In communication technology, terminated convolutional codes are mostly used in concatenation with other systematic or unsystematic block codes. In particular, the decoding result of a convolutional decoder is used as the input for another decoder.

[0053] To ensure the lowest possible error rate, it is necessary to supply "soft" decoding decisions instead of "hard" ones in the convolutional decoding for the further decoder, i.e., to generate a tuple of "soft" values (soft outputs) from R instead of a tuple of "hard" Boolean (±1) values. The absolute value of the respective "soft" decision then provides a safety measure for the correctness of the decision.

[0054] In principle, these soft outputs can be calculated in accordance with equation (1), depending on the channel model. However, the numeric complexity for calculating a soft output is $O(2^K)$, where K specifies the number of information bits. If K is realistically large, then these formulae can not be evaluated, in particular, since such a code word must be calculated again every few milliseconds (a real-time requirement).

[0055] One consequence of this is that soft outputs are dispensed with (with all consequences for the word and bit error rates) or, respectively, fewer elaborate approximations are performed for determining the soft outputs.

[0056] In the text which follows, a possibility for terminated convolutional codes is specified with the aid of which this complexity can be reduced to $O(K)$ in a trellis representation for calculating all soft outputs, i.e., this solution provides the possibility for a precise evaluation of equation (1).

[0057] In the text which follows, the bits of the code are represented in {±1} representation. In comparison with a {0, 1} representation, which is often used in information technology, -1 corresponds to 1 and 1 corresponds to 0.

[0058] On a body {±1}, addition $\oplus$ and multiplication $\odot$ are defined as follows:

$$-1 \oplus -1 = 1 \qquad -1 \odot -1 = -1$$

$$-1 \oplus 1 = -1 \qquad -1 \odot 1 = 1$$

$$1 \oplus -1 = -1 \qquad 1 \odot -1 = 1$$

$$1 \oplus 1 = 1 \qquad 1 \odot 1 = 1$$

[0059] The coding is done with the aid of a "shift register" into which bit blocks (input blocks) of the information bits are written with each clock pulse. The combination of the bits of the shift register then generates one bit block of the code word. The shift register is pre-assigned +1 bits in each case. To terminate the coding (termination) blocks of tail zeros (+1) are shifted in afterwards. As has been mentioned initially, check bits by way of which bit errors can be corrected are correlated with the information bits by way of coding.

[0060] The following are defined for the further embodiments:

[0061] $b \in \mathrm{N}$ number of input bits per block;

[0062] $V:=\{\pm 1\}^b$ set of state transition signs;

[0063] $a \in \mathrm{N}$ number of input blocks;

[0064] $K:= a \cdot b$ number of information bits without tail zeros ;

[0065] $k \in \mathrm{N}, k \geq 2$ block length of the shift register, penetration depth;

[0066] $L:= k \cdot b$ bit length of the shift register;

[0067] $S:=\{\pm 1\}^L$ set of shift register signs;

[0068] $n \in \mathrm{N}$ number of output bits per block;

[0069] $Q:= a + k - 1$ number of state transitions, input blocks + zeros;

[0070] $N:= n \cdot Q$ number of code bits; and

[0071] $R:=\frac{b}{n}$ code rate.

[0072] It should be noted here that the code rate is not K/N since the information bits have been counted without the zeros (+1) of the convolutional termination.

[0073] Furthermore, $s_0 \in S$ and $v_0 \in V$ are assumed to be the respective zero elements, i.e.,

[0074]
$$s_0 = (+1, \ldots, +1)^\top, \quad v_0 = (+1, \ldots, +1)^\top. \tag{9}$$

[0075] The state transition function of the shift register is assumed to be

$$T: \; S \times V \to S, \tag{10}$$

$$(s, v) \mapsto (s^{b+1}, \ldots, s^L, v^1, \ldots, v^b)^\top. \tag{11}$$

[0076]

[0077] The terminated convolutional code is defined by the characterizing subsets

$$M_1, \ldots, M_n \subseteq \{1, \ldots, L\}, \tag{12}$$

[0078]

[0079] (combination of register bits, alternatively in polynomial representation).

[0080] The current register content is coded via

$$C: \; S \to \{\pm 1\}^n, \tag{13}$$

$$s \mapsto C(s) \text{ where } C_j(s) := \bigoplus_{i \in M_j} s^i, \quad \text{for} \quad 1 \le j \le n. \tag{14}$$

[0081]

[0082] where s^i is the i-th component of s.

[0083] Finally, the coding of an information word is defined by way of

$$\varphi: \; \{\pm 1\}^K \to \{\pm 1\}^N, \tag{15}$$

$$u \mapsto \begin{pmatrix} C(s_1) \\ \vdots \\ (s_Q) \end{pmatrix}, \tag{16}$$

[0084]

where $s_0 \in S$ is the zero state (zero element),

$$u = \begin{pmatrix} \nu_1 \\ \vdots \\ \nu_a \end{pmatrix}, \quad \nu_i \in V, \quad 1 \le i \le a, \tag{17}$$

$$\nu_i := v_0, \quad a + 1 \le i \le Q, \tag{18}$$

[0085]

[0086] and furthermore

$$s_i := T(s_{i-1}, \nu_i), \quad 1 \le i \le Q. \tag{19}$$

[0087]

[0088] According to the definition of T, the following is obtained

$$s_{Q+1} := T(s_Q, v_0) = s_0. \tag{20}$$

[0089]

[0090] Accordingly, the set of all code words is

$$\varphi\left(\{\pm 1\}^K\right) := \left\{\varphi(u) \in \{\pm 1\}^N;\ u \in \{\pm 1\}^K\right\}. \tag{21}$$

[0091]

[0092] Often, polynomials

$$p_j \in \{0,1\}[D] \quad \text{where} \quad \deg(p_j) \leq L-1$$

[0093]

[0094] are used instead of the sets M_j for code definition, i.e.,

$$p_j(D) = \sum_{i=0}^{L-1} \gamma_{i,j} D^i, \tag{22}$$

[0095]

[0096] with

$$\gamma_{i,j} \in \{0,1\} \qquad \begin{aligned} i &= 0, \ldots, L-1, \\ j &= 1, \ldots, n. \end{aligned}$$

[0097]

[0098] The following transformations then apply for $j = 1, \ldots, n$:

$$M_j = \{i \in \{1, \ldots, L\};\ \gamma_{L-i,j} = 1\} \tag{23}$$

$$p_j(D) = \sum_{i \in M_j} D^{L-i}. \tag{24}$$

[0099]

BLOCK CODE REPRESENTATION

[00100] Since a terminated convolutional code is a block code, the code bits c_j, $1 \leq j \leq N$ can also be represented from the information bits u_i, $1 \leq i \leq K$, with index sets J_j, as follows:

[00101]
$$c_j := \bigoplus_{i \in J_j} u_i, \quad \text{for } 1 \le j \le \dot{N}, \tag{25}$$

[00102] where

[00103]
$$J_1, \ldots, J_N \subseteq \{1, \ldots, K\}. \tag{26}$$

[00104] The index sets J_j can be calculated directly from the above index sets M_m of the code definition.

[00105] Consider

$$j = n(q-1) + m, \quad q = 1, \ldots, Q, \; m = 1, \ldots, n. \tag{27}$$

[00106]
$$c_j = C_m(s_q) = \bigoplus_{i \in M_m} (s_q)^i = \bigoplus_{i \in M_m} u_{i+b(q-k)}, \tag{28}$$

[00107] where $u_i := +1$ for $i \notin \{1,\ldots,K\}$.

[00108] Furthermore,

[00109]
$$c_j = \bigoplus_{i-b(q-k) \in M_m} u_i = \bigoplus_{i \in M_m + b(q-k)} u_i, \tag{29}$$

[00110] and it thus follows for $j = 1,\ldots,N$ that

[00111]
$$\begin{aligned} J_j &= \{1, \ldots, K\} \cap (M_m + b(q-k)) \\ &= \{i \in \{1, \ldots, K\};\; i - b(q-k) \in M_m\}. \end{aligned} \tag{30}$$

EXAMPLE: SACCH CONVOLUTIONAL CODE

[00112] In the above terminology, the convolutional code described in section 4.1.3 of the GSM Technical Specification GSM 05.03, Version 5.2.0 (channel coding) is:

[00113] $b = 1$ number of input bits per block;

[00114] $V=\{\pm 1\}$ set of state transition signs;

[00115] $a = 224$ number of input blocks;

[00116] $K = 224$ number of information bits without tail zeros;

[00117] $k = 5$ block length of the shift register, depth of penetration;

[00118] $L = 5$ bit length of the shift register;

[00119] $S = \{\pm 1\}^5$ set of shift register signs;

[00120] $n = 2$ number of output bits per block;

[00121] $Q = 228$ number of state transitions, input blocks + zeros;

[00122] $N = 456$ number of code bits;

[00123] $R = \frac{1}{2}$ code rate;

[00124] $M_1 = \{1,2,5\}$ characterizing set; polynomial: $1+D^3+D^4$; and

[00125] $M_2 = \{1,2,4,5\}$ characterizing set; polynomial: $1+D+D^3+D^4$.

SOFT OUTPUTS IN AN AWGN CHANNEL MODEL

[00126] In the text which follows, calculation rules for determining the soft outputs are derived, especially for the sake of clarity.

[00127] For this purpose, a probability space (Ω, S, P) and a K-dimensional random variable $U{:}\Omega \rightarrow \{\pm 1\}^K$ are considered which have the properties

- The components $U_1,\ldots,U_K : \Omega \rightarrow \{\pm 1\}$ are stochastically independent.
- The following holds for $i = 1,\ldots,K$

[00128]
$$P(\{\omega \in \Omega;\ U_i(\omega) = -1\}) = P(\{\omega \in \Omega;\ U_i(\omega) = +1\}). \tag{31}$$

[00129] Figure 1 shows a representation of digital telecommunication. A unit consisting of source 201, source encoder 202 and crypto-encoder 203 determines an information item $u \in \{\pm 1\}^K$ which is used as input for one (or possibly more) channel encoder(s) 204. The channel encoder 204 generates a code word $c \in \{\pm 1\}^N$ which is fed into a modulator 205 and is transmitted via a disturbed physical channel 206 to a receiver where it is determined to become a real-value code word $y \in \mathrm{R}^N$ in a demodulator 207. This code word is converted into a real-value information item in a channel decoder 208. If necessary, a "hard" correlation with the Boolean values ±1 can also be made in a further decoder so that

the received information is present in Boolean notation. The receiver is completed by a unit of crypto-decoder 209, source decoder 210 and sink 211. The two crypto-encoder 203 and crypto-decoder 209 units are optional in this arrangement.

[00130] The information to be reconstructed, $u \in \{\pm 1\}^K$, of the crypto-encoder 203 is interpreted as implementation of the random variables U since nothing is known about the choice of u in the receiver.

[00131] Thus, the output $c \in \{\pm 1\}^N$ of the channel encoder 204 is an implementation of the random variables $\varphi(U)$.

[00132] The output $y \in \mathrm{R}^N$ of the demodulator 207 is interpreted as implementation of the random variables

$$Y : \Omega \rightarrow \mathbb{R}^N, \tag{32}$$

$$\omega \mapsto \varphi(U(\omega)) + Z(\omega), \tag{33}$$

[00133]

[00134] a random variable $Z : \Omega \rightarrow \mathrm{R}^N$ representing the channel disturbances in the physical channel 206.

[00135] In the text which follows, an AWGN channel model is assumed, i.e., Z is a $N(0,\sigma^2 I_N)$ normally distributed random variable which is stochastically independent of U and, respectively, $\varphi(U)$. The variance σ^2 is calculated from the ratio between noise power density and mean energy in the channel 206 and is here assumed to be known.

[00136] The unknown output $u \in \{\pm 1\}^K$ of the crypto-encoder is to be reconstructed on the basis of an implementation y of Y. To estimate the unknown quantities $u_1,\ldots,u_K$, the distribution of the random variables U is investigated given the condition that y has been received.

[00137] The consequence of the fact that the random variable Y is a steady random variable is that the consideration of U under the condition that y has been received ($Y(\hat{\omega})=y$) is extremely complicated.

[00138] Firstly, the following is defined for $i \in \{1,\ldots,K\}$ and $\alpha \in \{\pm 1\}$

$$\Gamma^i(\alpha) := \left\{\varphi(u);\ u \in \{\pm 1\}^K;\ u_i = \alpha\right\}. \tag{34}$$

[00139]

[00140] In a preparatory step, the following quantities are considered for $\epsilon > 0$, paying attention to the injectivity of the coding map φ:

$$L_\epsilon(U_i|y) := \ln\left(\frac{P(\{\omega\in\Omega;\ U_i(\omega)=+1\}\mid\{\omega\in\Omega;\ Y(\omega)\in M_{y,\epsilon}\})}{P(\{\omega\in\Omega;\ U_i(\omega)=-1\}\mid\{\omega\in\Omega;\ Y(\omega)\in M_{y,\epsilon}\})}\right)$$
$$= \ln\left(\frac{\sum_{c\in\Gamma^i(+1)} P(\{\omega\in\Omega;\ \varphi(U(\omega))=c\}\mid\{\omega\in\Omega;\ Y(\omega)\in M_{y,\epsilon}\})}{\sum_{c\in\Gamma^i(-1)} P(\{\omega\in\Omega;\ \varphi(U(\omega))=c\}\mid\{\omega\in\Omega;\ Y(\omega)\in M_{y,\epsilon}\})}\right), \tag{35}$$

[00141]

[00142] for $i=1,\ldots,K$, where $M_{y,\varepsilon}:=[y_1,y_1+\varepsilon]\times\ldots\times[y_N,y_N+\varepsilon]$.

[00143] Using the theorem by Bayes, the following is obtained:

$$L_\epsilon(U_i|y) = \ln\left(\frac{\sum_{c\in\Gamma^i(+1)} P(\{\omega\in\Omega;\ Y(\omega)\in M_{y,\epsilon}\}\mid\{\omega\in\Omega;\ \varphi(U(\omega))=c\})}{\sum_{c\in\Gamma^i(-1)} P(\{\omega\in\Omega;\ Y(\omega)\in M_{y,\epsilon}\}\mid\{\omega\in\Omega;\ \varphi(U(\omega))=c\})}\right)$$
$$= \ln\left(\frac{\sum_{c\in\Gamma^i(+1)}\int_{M_{y,\epsilon}} \exp\left(-\frac{(x-c)^T(x-c)}{2\sigma^2}\right)dx}{\sum_{c\in\Gamma^i(-1)}\int_{M_{y,\epsilon}} \exp\left(-\frac{(x-c)^T(x-c)}{2\sigma^2}\right)dx}\right). \tag{36}$$

[00144]

[00145] Considering then the limiting process of $L_\varepsilon(U_i|y)$ for $\varepsilon \downarrow 0$ by using L'Hospital's rule several times, the soft output $L(U_i|y)$ is obtained for each symbol as in equation (1).

[00146] Since

$$\Gamma^i(+1)\cup\Gamma^i(-1)=\{\pm1\}^K$$

[00147]

[00148] holds true, a total of $O(2^K)$ numeric operations are necessary for evaluating equation (1).

[00149] The vector $L(U_\bullet|y) \in \mathbb{R}^K$ is the result of decoder 208.

REDUCTION OF COMPLEXITY IN THE DETERMINATION OF THE SOFT OUTPUTS

SOFT-OUTPUT DETERMINATION FOR CONVOLUTIONAL CODES

[00150] Firstly, the special characteristics of the terminated convolutional coding are used for providing an organized representation of the soft-output formula (1).

[00151] For an arbitrary, but preselected output $y \in \mathrm{R}^N$ of the demodulator 207, the following weighting function (a Viterbi metric) of code words is considered:

$$F:\ \{\pm 1\}^N \to \mathbb{R}_0^+, \tag{37}$$

$$c \mapsto \sum_{j=1}^{N} (y_j - c_j)^2. \tag{38}$$

[00152]

[00153] For permissible code words $c \in \{\pm 1\}^N$, i.e. $c \in \varphi(\{\pm 1\}^K)$, $F(c)$ can be reduced as follows, using the shift register representation:

$$F(c) = \sum_{q=1}^{Q} \underbrace{\sum_{j=1}^{n} \left(y_{n(q-1)+j} - C_j(\tilde{s}_q^c)\right)^2}_{=:\Delta F_q(\tilde{s}_q^c)}, \tag{39}$$

[00154]

[00155] where $\tilde{s}_q^c$ stands for the q-th state of the shift register in the (unambiguous) generation of the word c.

[00156] Then the following is defined for I=1,...,K and $\alpha \in \{\pm 1\}$:

$$A_\alpha^i(y) := \sum_{c \in \Gamma^i(\alpha)} \exp\left(-\frac{(y-c)^\top (y-c)}{2\sigma^2}\right) = \sum_{c \in \Gamma^i(\alpha)} \prod_{q=1}^{Q} \exp\left(-\frac{1}{2\sigma^2} \Delta F_q(\tilde{s}_q^c)\right). \tag{40}$$

[00157]

[00158] Thus, the following holds true for the soft outputs

$$L(U_i|y) = \ln\left(\frac{A_{+1}^i(y)}{A_{-1}^i(y)}\right), \quad i = 1, \ldots, K. \tag{41}$$

[00159]

[00160] In the text which follows, the values $A_\alpha^i(y)$ are determined with the aid of a trellis diagram / representation.

[00161] To reduce the complexity of calculation, the following procedure is adopted in the following sections:

[00162] • generalization of A_α^i by mappings $\tilde{A}_m$.

[00163] • Recursive representation of $\widetilde{A}_m$ by mappings A_m, the values of which are calculated with a "from left to right" run through a trellis diagram.

[00164] • Reversal of the recursion by mappings B_m, the values of which are calculated with a "from right to left" run through a trellis diagram.

[00165] • joint calculation of all A^i_α by way of a further run through a trellis diagram by using A_m and B_m.

[00166] The trellis diagram is here a set

$$\mathcal{T} = \{(s, q);\ s \in S,\ q = 0, \ldots, Q+1\} \tag{42}$$

[00167]

[00168] The elements (s,q) of this set are also called the nodes in the trellis diagram, s representing a state and q being considered as a dynamic value (especially time).

GENERAL RECURSIVE REPRESENTATION

[00169] Firstly, some definitions are needed for representing the A^i_α in a generalized form which allows later transformation. For this reason, the following is determined

$$s_1^u := T(s_0, u_1), \quad u \in V^m = V \times \ldots \times V,\ m \geq 1, \tag{43}$$
$$s_j^u := T(s_{j-1}^u, u_j) \quad u \in V^m,\ m \geq j > 2, \tag{44}$$

[00170]

[00171] i.e., s_j^u represents the state of the shift register after j shifts of the register with the input symbols $u_1,\ldots,u_j$.

[00172] Furthermore, sets $V_j \subseteq V, j \in$ N, which contain the permissible state transition symbols in the j-th step, are considered. Furthermore, product sets are defined as

$$U_m := V_1 \times \ldots \times V_m \subseteq V^m, \quad m \in \mathbb{N}, \tag{45}$$

[00173]

[00174] i.e., U_m contains the first m components of the permissible input words.

[00175] For $q \in$ N, mappings

$$\mu_q : S \to \mathbb{R} \tag{46}$$

[00176]

[00177] are considered and for $m \in$ N and input word sets $U_m \subseteq V^m$, mappings are defined as follows

$$\tilde{A}_m : \wp(S) \to \mathbb{R}, \tag{47}$$

$$E \mapsto \sum_{\substack{(u \in U_m) \\ \wedge (s_m^u \in E)}} \prod_{j=1}^{m} \mu_j(s_j^u), \tag{48}$$

[00178]

[00179] i.e., summing over all permissible input words, the shift register of which reaches a final state in E, is performed. If there are no such input words, the sum is determined as 0 over an empty index set.

[00180] In addition, a mapping is determined as

$$W : S \times \wp(V) \to \wp(S), \tag{49}$$

$$(t, \hat{V}) \mapsto \left\{ s \in S;\ \exists \hat{v} \in \hat{V} \ni T(s, \hat{v}) = t \right\}, \tag{50}$$

[00181]

[00182] i.e., W maps $(t, \hat{V})$ into the sets of all states which can reach the state t with a transition symbol from $\hat{V}$.

[00183] The following holds true for $m \geq 2$, $E \subseteq S$

$$
\begin{aligned}
\tilde{A}_m(E) &= \sum_{\substack{(u \in U_m) \\ \wedge (s_m^u \in E)}} \prod_{j=1}^{m} \mu_j(s_j^u) \\
&= \sum_{s \in E} \sum_{\substack{(u \in U_m) \\ \wedge (s_m^u = s)}} \prod_{j=1}^{m} \mu_j(s_j^u) \\
&= \sum_{s \in E} \mu_m(s) \sum_{\substack{(u \in U_m) \\ \wedge (s_m^u = s)}} \prod_{j=1}^{m-1} \mu_j(s_j^u) \\
&= \sum_{s \in E} \mu_m(s) \sum_{\substack{(u \in U_{m-1}) \\ \wedge (s_{m-1}^u \in W(s, V_m))}} \prod_{j=1}^{m-1} \mu_j(s_j^u) \\
&= \sum_{s \in E} \mu_m(s) \tilde{A}_{m-1}\left(W(s, V_m)\right). \qquad (51)
\end{aligned}
$$

[00184]

[00185] In the transformation in the last step but one, attention must be paid to the fact that there is **exactly one** transition symbol $v \in V_m$ with $T\left(s_{m-1}^u, v\right) = s$, if s_{m-1}^u is in $W(s, V_m)$, i.e., it is not necessary to take account of any multiplicities.

[00186] Consider, then, the following for $m \geq 2$ mappings

$$
\begin{aligned}
A_m &: S \rightarrow \mathbb{R}, \qquad (52) \\
&s \mapsto \mu_m(s) \tilde{A}_{m-1}\left(W(s, V_m)\right). \qquad (53)
\end{aligned}
$$

[00187]

[00188] Thus, a recursion formula can be derived for $m \geq 3$:

$$
\begin{aligned}
A_m(s) &= \mu_m(s) \tilde{A}_{m-1}\left(W(s, V_m)\right) \\
&= \mu_m(s) \sum_{t \in W(s, V_m)} \mu_{m-1}(t) \tilde{A}_{m-2}\left(W(t, V_{m-1})\right) \\
&= \mu_m(s) \sum_{t \in W(s, V_m)} A_{m-1}(t). \qquad (54)
\end{aligned}
$$

[00189]

[00190] Furthermore:

$$\begin{aligned} A_2(s) &= \mu_2(s)\tilde{A}_1\left(W(s, V_2)\right) \\ &= \mu_2(s) \sum_{\substack{(u \in U_1) \\ \wedge (s_1^u \in W(s, V_2))}} \mu_1(s_1^u) \\ &= \mu_2(s) \sum_{t \in W(s, V_2)} \mu_1(t)\delta_{s_0 \in W(t, V_1)} \\ &= \mu_2(s) \sum_{t \in W(s, V_2)} \underbrace{\mu_1(t) \sum_{\hat{t} \in W(t, V_1)} \underbrace{\delta_{\hat{t}=s_0}}_{=:A_0(\hat{t})}}_{=:A_1(t)} \end{aligned} \tag{55}$$

[00191]

[00192] In summary, the following thus holds true for $s \in S$, $E \subseteq S$:

$$A_0(s) = \begin{cases} 1, & \text{for} \quad s = s_0, \\ 0, & \text{otherwise} \end{cases}, \tag{56}$$

$$A_m(s) = \mu_m(s) \sum_{t \in W(s, V_m)} A_{m-1}(t), \quad \text{for} \quad m \in \mathbb{N}, \tag{57}$$

$$\tilde{A}_m(E) = \sum_{s \in E} A_m(s), \quad \text{for} \quad m \in \mathbb{N}. \tag{58}$$

[00193]

[00194] The sets $W(s, V_m)$ can be represented constructively. For this purpose, two further mappings are considered. The following is defined

$$\tau : S \to V, \tag{59}$$

$$s = (s^1, \ldots, s^L)^\top \mapsto (s^{L-b+1}, \ldots, s^L)^\top, \tag{60}$$

[00195]

[00196] i.e., if the state s is the result of a state transition, then $\tau(s)$ was the associated state transition symbol.

[00197] Furthermore

$$\hat{T} : V \times S \to S, \tag{61}$$

$$(v, s) \mapsto (v^1, \ldots, v^b, s^1, \ldots, s^{L-b})^\top, \tag{62}$$

[00198]

[00199] is defined, i.e., $\hat{T}$ reverses the direction of the shift register operation.

[00200] The following then holds

$$T\left(\hat{T}(v,s),\tau(s)\right)=s, \quad \text{for all} \quad s\in S,\ v\in V \tag{63}$$

[00201]

[00202] and for all $t \in S$ and $\hat{V} \subseteq V$, it also holds true that

$$\begin{aligned} W(t,\hat{V}) &= \left\{s\in S;\ \exists\hat{v}\in\hat{V}\ni T(s,\hat{v})=t\right\} \\ &= \begin{cases} \left\{\hat{T}(v,t);\ v\in V\right\}, & \text{if } \tau(t)\in\hat{V}, \\ \emptyset, & \text{else.} \end{cases} \end{aligned} \tag{64}$$

[00203]

[00204] Thus, the recursion formula (57) for $A_m(s)$ can be written down constructively as follows:

$$\begin{aligned} A_m(s) &= \mu_m(s) \sum_{t\in W(s,V_m)} A_{m-1}(t) \\ &= \begin{cases} \mu_m(s)\sum_{v\in V} A_{m-1}\left(\hat{T}(v,s)\right), & \text{if } \tau(s)\in V_m, \\ 0, & \text{else.} \end{cases} \end{aligned} \tag{65}$$

[00205]

[00206] It should be noted that in this section, no restrictions were set for the set V of the state transition symbols and for the sets $V_j \in \wp(V)$.

REVERSAL OF RECURSION

[00207] In the text which follows, a recursion in the "reverse direction" compared with the above recursion is described. This new recursion is defined with the aid of the recursion formula (57) for $A_m(s)$.

[00208] The following is assumed for this purpose

$$T(t,\hat{V}) := \left\{T(t,\hat{v});\ \hat{v}\in\hat{V}\right\}, \quad \text{for} \quad t\in S,\ \hat{V}\subseteq V \tag{66}$$

[00209]

[00210] and for $M \in$ N, $0 \le m \le Q$, the following mappings are considered

$$B_m : S \to \mathbb{R}, \tag{67}$$

[00211]

[00212] with the following recursive characteristic:

$$\sum_{s\in S} A_m(s) \sum_{t\in T(s,V_{m+1})} B_{Q-m}(t) =$$
$$= \sum_{s\in S} \mu_m(s) \sum_{\hat{t}\in W(s,V_m)} A_{m-1}(\hat{t}) \sum_{t\in T(s,V_{m+1})} B_{Q-m}(t)$$
$$= \sum_{\hat{t}\in S} \sum_{s\in T(\hat{t},V_m)} \mu_m(s) A_{m-1}(\hat{t}) \sum_{t\in T(s,V_{m+1})} B_{Q-m}(t)$$
$$= \sum_{\hat{t}\in S} A_{m-1}(\hat{t}) \sum_{s\in T(\hat{t},V_m)} \underbrace{\mu_m(s) \sum_{t\in T(s,V_{m+1})} B_{Q-m}(t)}_{=:B_{Q-m+1}(s)},$$

[00213]

[00214] i.e.,

$$\sum_{s\in S} A_m(s) \sum_{t\in T(s,V_{m+1})} B_{Q-m}(t) = \sum_{s\in S} A_{m-1}(s) \sum_{t\in T(s,V_m)} B_{Q-m+1}(t). \qquad (68)$$

[00215]

[00216] By applying equation (68) several times, the following is obtained for an arbitrary $j \in \{1,\ldots,m+1\}$

$$\sum_{s\in S} A_m(s) \sum_{t\in T(s,V_{m+1})} B_{Q-m}(t) = \sum_{s\in S} A_{j-1}(s) \sum_{t\in T(s,V_j)} B_{Q-j+1}(t). \qquad (69)$$

[00217]

[00218] According to the above definition, the recursion formula is thus

$$B_m(s) = \mu_{Q-m+1}(s) \sum_{t\in T(s,V_{Q-m+2})} B_{m-1}(t), \qquad \text{for } 1 \le m \le Q. \qquad (70)$$

[00219]

[00220] To terminate the recursion, the following are defined

$$B_0(s) = \begin{cases} 1, & \text{for } s = s_0, \\ 0, & \text{else} \end{cases}. \qquad (71)$$

[00221]

[00222] Given this termination and the equations (58) and (69),

$$\tilde{A}_Q(W(s_0, V_{Q+1}))$$

[00223]

[00224] can be represented for $V_{Q+1} := \{v_0\}$ and with an arbitrary $j \in \{1,...,Q+1\}$ as follows:

$$\begin{aligned}
\tilde{A}_Q(W(s_0, V_{Q+1})) &= \sum_{s \in W(s_0, V_{Q+1})} A_Q(s) \\
&= \sum_{s \in S} A_Q(s) \sum_{t \in T(s,\{v_0\})} B_0(t) \\
&= \sum_{s \in S} A_Q(s) \sum_{t \in T(s,V_{Q+1})} B_0(t) \\
&= \sum_{s \in S} A_{j-1}(s) \sum_{t \in T(s,V_j)} B_{Q-j+1}(t). \qquad (72)
\end{aligned}$$

[00225]

[00226] Note: in the evaluation of (72), V_j is not included in the calculation of the A_m and B_m needed.

CALCULATION OF A_α^i

[00227] Using the preliminary work from the preceding sections, A_α^i can now be calculated in a simple manner.

[00228] For this purpose, the following are defined:

$$V_j := V, \quad \text{for} \quad j \in \{1, \ldots, a\}, \qquad (73)$$
$$V_j := \{v_0\}, \quad \text{for} \quad j \in \{a+1, \ldots, Q+1\}, \qquad (74)$$

[00229]

[00230] i.e., all permissible code words are defined via the states s_j^u with

$$u \in U_Q = V_1 \times \ldots \times V_Q$$

[00231]

[00232] The code words used in the calculation of the A_α^i are restricted by u_i=α. For an arbitrary but fixed choice of $i \in \{1,...,K\}$, there is exactly one $j \in \{1,...,a\}$ and exactly one $\hat{i} \in \{1,...,b\}$ with

$$i = (j-1) \cdot b + \hat{i}. \tag{75}$$

[00233]

[00234] Furthermore, the following are defined for an arbitrary but fixed choice of $\alpha \in \{\pm 1\}$:

$$V_j^i(\alpha) := \{v \in V;\ v_{\hat{i}} = \alpha\} \tag{76}$$

$$U_Q^i(\alpha) := V_1 \times \ldots \times V_{j-1} \times V_j^i(\alpha) \times V_{j+1} \times \ldots \times V_Q \subset U_Q, \tag{77}$$

[00235]

i.e., the code words from $\Gamma^i(\alpha)$ are determined via the states s_j^u with $u \in U_Q^i(\alpha)$.

[00236] For an arbitrary, but fixed choice of $y \in \mathrm{R}^N$, define for $q \in \{1,...,Q\}$

$$\mu_q : S \to \mathbb{R}, \tag{78}$$

$$s \mapsto \exp\left(-\frac{1}{2\sigma^2}\sum_{j=1}^{n}\left(y_{n(q-1)+j} - C_j(s)\right)^2\right) = \exp\left(-\frac{1}{2\sigma^2}\Delta F_q(s)\right). \tag{79}$$

[00237]

[00238] According to the definition of the convolutional code, the following holds true for all s_Q^u with $u \in U_Q$:

$$s_{Q+1}^u = T(s_Q^u, u_{Q+1}) = s_0, \quad u_{Q+1} \in V_{Q+1} = \{v_0\}, \tag{80}$$

[00239]

[00240] i.e.,

$$s_Q^u \in W(s_0, V_{Q+1}). \tag{81}$$

[00241]

[00242] Taking account of equation (72), the following thus holds true:

$$
\begin{aligned}
A_\alpha^i(y) &= \sum_{c\in\Gamma^i(\alpha)} \prod_{q=1}^{Q} \exp\left(-\frac{1}{2\sigma^2}\Delta F_q(\tilde{s}_q^c)\right) \\
&= \sum_{u\in U_Q^i(\alpha)} \prod_{q=1}^{Q} \mu_q(s_q^u) \\
&= \sum_{\substack{(u\in U_Q^i(\alpha)) \\ \wedge(s_Q^u\in W(s_0,V_{Q+1}))}} \prod_{q=1}^{Q} \mu_q(s_q^u) \\
&= \tilde{A}_Q(W(s_0, V_{Q+1})) \\
&= \sum_{s\in S} A_{j-1}(s) \sum_{t\in T(s,V_j^i(\alpha))} B_{Q-j+1}(t) \qquad (82)
\end{aligned}
$$

[00243]

[00244] The important factor is that the A_m and B_m needed can be calculated independently of i and α via U_Q and, respectively, U_{Q+1}. Above, $\tilde{A}_Q(W(s_0,V_{Q+1}))$ was formally determined via the auxiliary construct $U_Q^i(\alpha)$ which, however, is no longer needed in the resultant explicit representation.

SUMMARY OF THE PROCEDURE:

- Define

$$
\begin{aligned}
V_j &:= V, && \text{for } j\in\{1,\ldots,a\}, \\
V_j &:= \{v_0\}, && \text{for } j\in\{a+1,\ldots,Q+1\}, \\
V_j^i(\alpha) &:= \{v\in V;\ v_{\hat{i}}=\alpha\}, && \text{for } i=(j-1)\cdot b+\hat{i}, \\
& && \hat{i}\in\{1,\ldots,b\}, \\
& && j\in\{1,\ldots,a\},\ \alpha\in\{\pm 1\}.
\end{aligned}
$$

[00245]

- For an arbitrary, but fixed choice of $y \in \mathrm{R}^N$, define for $q \in \{1,\ldots,Q\}$

$$
\begin{aligned}
&\mu_q : S \to \mathbb{R}, \\
&s \mapsto \exp\left(-\frac{1}{2\sigma^2}\sum_{j=1}^{n}\left(y_{n(q-1)+j} - C_j(s)\right)^2\right) = \exp\left(-\frac{1}{2\sigma^2}\Delta F_q(s)\right).
\end{aligned}
$$

[00246]

- Calculate

$$A_m(s), \quad \text{for} \quad s \in S, \; m \in \{1, \ldots, a-1\},$$
$$B_m(s), \quad \text{for} \quad s \in S, \; m \in \{1, \ldots, Q\},$$

[00247]

[00248] according to the recursion formulae (57) and (70) and starting values $A_0(s)$, $B_0(s)$, specified above, with (56) and (71).

- Calculate all A_α^i, $i \in \{1,...,L\}$, $\alpha \in \{\pm 1\}$ over

$$A_\alpha^i(y) = \sum_{s \in S} A_{j-1}(s) \sum_{t \in T(s, V_j^i(\alpha))} B_{Q-j+1}(t). \tag{83}$$

[00249]

[00250] and determine the soft outputs

$$L(U_i|y) = \ln\left(\frac{A_{+1}^i(y)}{A_{-1}^i(y)}\right), \quad i = 1, \ldots, K.$$

[00251]

[00252] Together with the recursion formula from the preceding section, all $A_\alpha^i(y)$ can now be calculated jointly with $O(2^L \cdot Q)$ or, respectively, $O(K)$ operations instead of $O(K2^K)$ operations.

[00253] Note that

[00254] $L = k \cdot b$, $Q = a + k - 1$, $K = a \cdot b$,

[00255] where a is the number of information bits.

[00256] The numeric complexity for calculating the soft outputs has thus been reduced from an exponential order to a linear order where a, the number of information bits, is the decisive quantity.

SPECIAL CASE: BINARY STATE TRANSITION ($B = 1$)

[00257] In the important special case of $b = 1$, the set V of state transition symbols only consists of the two elements +1, -1. The GSM codes, for instance, belong to this widespread special case.

[00258] Since now $i = j$ and $V_j^i(\alpha) = \{\alpha\}$ in the above description, the procedure is simplified as follows:

- Define

$$V_j := \{\pm 1\}, \quad \text{for } j \in \{1, \ldots, a\},$$
$$V_j := \{+1\}, \quad \text{for } j \in \{a+1, \ldots, Q+1\}$$

[00259]

- For an arbitrary, but fixed choice of $y \in R^N$ define for $q \in \{1,\ldots,Q\}$

$$\mu_q : S \to \mathbb{R},$$
$$s \mapsto \exp\left(-\frac{1}{2\sigma^2}\sum_{j=1}^{n}\left(y_{n(q-1)+j} - C_j(s)\right)^2\right) = \exp\left(-\frac{1}{2\sigma^2}\Delta F_q(s)\right).$$

[00260]

- Calculate

$$A_m(s), \quad \text{for } s \in S, \; m \in \{1, \ldots, a-1\},$$
$$B_m(s), \quad \text{for } s \in S, \; m \in \{1, \ldots, Q\},$$

[00261]

[00262] according to the recursion formulae (57) and (70) and starting values $A_0(s)$, $B_0(s)$ with (56) and (71).

- Calculate all A_α^i, $i \in \{1,\ldots,K\}$, $\alpha \in \{\pm 1\}$ over

$$A_\alpha^i(y) = \sum_{s \in S} A_{i-1}(s) B_{Q-i+1}(T(s,\alpha)). \tag{84}$$

[00263]

[00264] and determine the soft outputs

$$L(U_i|y) = \ln\left(\frac{A_{+1}^i(y)}{A_{-1}^i(y)}\right), \quad i = 1, \ldots, K.$$

[00265]

ALGORITHMIC CONVERSION

[00266] For the algorithmic conversion, consider the trellis diagram

$$\mathcal{T} = \{(s, q);\ s \in S,\ q = 0, \ldots, Q + 1\}$$

[00267]

[00268] and the mappings

- node weights in state s of trellis segment q

$$\mu:\ \mathcal{T} \rightarrow \mathbb{R},$$
$$(s, q) \mapsto \exp\left(-\frac{1}{2\sigma^2}\Delta F_q(s)\right)$$

[00269]

- Subtotals 'A' in state s of trellis segment q

$$A:\ \mathcal{T} \rightarrow \mathbb{R},$$
$$(s, q) \mapsto A(s, q)$$

[00270]

- Subtotals 'B' in state s of trellis segment Q-q+1

$$B:\ \mathcal{T} \rightarrow \mathbb{R},$$
$$(s, q) \mapsto B(s, q)$$

[00271]

[00272] The mappings are only evaluated in the meaningful subsets of the definition domain.

[00273] Figure 2 shows an algorithm in pseudocode notation which represents a progression in the trellis diagram, considering all states for the calculation of the node weights. The algorithm illustrates the above statements and is comprehensible out of itself. Since the value of $\Delta F_q(s)$ depends only indirectly on the state s and is formed directly with $C(s)$, the following holds true

$$|\{\Delta F_q(s);\ s \in S\}| \leq \min\{2^L, 2^n\},$$

[00274]

[00275] i.e., for $n < L$, many of the above $\mu(s,q)$ have the same value. Depending on the special code, $\mu(s,q)$ can thus be determined with far fewer operations in the implementation.

[00276] Figure 3 and Figure 4 each show an algorithm in pseudocode notation for determining soft outputs. Figure 3 relates to the general case and Figure 4 relates to the special case for the binary state transition ($b = 1$). Both algorithms illustrate the above statements and are comprehensible in and of themselves.

[00277] With a suitable implementation representation of V and, respectively, $V_j^{i}(\alpha)$, for instance as subsets of N, the above iterations $v \in V$ and $s \in S$ can be implemented as normal program loops. Naturally, indices which may occur such as, for example, $k - 1 + q$, are calculated only once in the implementation and not with every occurrence as is written down here for better clarity.

[00278] Figure 5 shows a processor unit PRZE. The processor unit PRZE comprises a processor CPU, a memory SPE and an input/output interface IOS which is used in various ways via an interface IFC: an output can be displayed on a monitor MON and/or output on a printer PRT via a graphics interface. An input is made via a mouse MAS or a keyboard TAST. The processor unit PRZE also has a data bus BUS which ensures the connection of a memory MEM, the processor CPU and the input/output interface IOS. Furthermore, additional components, for example an additional memory, data store (hard disk) or scanner, can be connected to the data bus BUS.

[00279] The above-described method and apparatus are illustrative of the principles of the present invention. Numerous modifications and adaptations will be readily apparent to those skilled in this art without departing from the spirit and scope of the present invention.

ABSTRACT

METHOD AND ARRANGEMENT FOR DECODING A PREDETERMINED CODE WORD

[00280] A method for decoding a predetermined code word is specified in which the code word comprises a number of positions having different values. In this method, encoding is performed, in particular, by way of a terminated convolutional code. Each position of the code word is correlated with a safety measure (soft output) for a most probable Boolean value by performing the correlation on the basis of a trellis representation. The decoding of the code word is determined by the correlation of the individual positions of the code word.

[~~DESCRIPTION~~]**SPECIFICATION**

TITLE

METHOD AND ARRANGEMENT FOR DECODING A PREDETERMINED CODE WORD

BACKGROUND OF THE INVENTION

FIELD OF THE INVENTION

[0001] The invention relates to a method and an arrangement for decoding a predetermined code word.

DESCRIPTION OF THE RELATED ART

[0002] In the decoding of a code word which has a predetermined number of positions, the information-carrying positions are[~~-to be~~] restored as completely as possible.

[0003] The decoding takes place at the end of the receiver which has received the code word via a disturbed channel. Signals are transmitted, in particular as Boolean values, preferably subdivided into +1 and -1, via the channel where they are subject to a disturbance, and are converted into analog values which can deviate to a greater or lesser extent from the predetermined Boolean values (±1) by a demodulator.

[0004] **The general assumption is K positions of binary information ("information bits") without redundancy $u \in \{\pm 1\}^K$, which is mapped into a code word c** [~~The general assumption is K positions of binary information ("information bits") without redundancy u~~][~~, which is mapped into a code word c~~] $\in \{\pm 1\}^N$ by means of systematic block codes or unsystematic block codes by a channel coder. In this arrangement, the code word contains $N - K$ bits (also "check bits") which can be used as redundant information to the N information bits for restoring the information after transmission via the disturbed channel.

[0005] The systematic block code adds to the N information bits $N - K$ check bits which are calculated from the information bits, the information bits themselves remaining unchanged whereas, in the unsystematic block code, the information bits themselves are changed, for example the information is in an operation performed from one to the next position. Here, too, check bits are provided for reconstructing the information hidden in the operations. In the text which follows, in particular, a technically significant variant of unsystematic block codes, the so-called terminated convolutional codes, is considered.

[0006] "Hard" decoding of a correlation of the received code word (with the positions occupied by analog values) i.e., correlating each position with the nearest Boolean value in each case, is a decisive disadvantage since valuable information is lost in this process.

SUMMARY OF THE INVENTION

[0007] It is the object of the invention to determine a decoding of a predetermined code word, with decoding supplying analog values (so-called "soft outputs") which, in particular, can be taken into consideration in the subsequent decoding method and thus provide for high error correction in the transmission of code words via a disturbed channel.

[0008] This object is achieved in accordance with the [~~features of the independent patent claims. Further developments of the invention are also obtained from the dependent claims~~]**various embodiments of the method and apparatus discussed below**.

[0009] To achieve [~~the~~]**this** object, a method for decoding a predetermined code word is specified in which the code word comprises a number of positions having different values. In this arrangement, encoding has taken place, in particular, by [~~means~~]**way** of a terminated convolutional code. Each position of the code word is correlated with a safety measure (soft output) for a most probable Boolean value by performing the correlation on the basis of a trellis representation. The decoding of the code word is determined by the correlation of the individual positions of the code word.

[0010] A decisive advantage [~~is~~]here **is** that due to the correlation based on the trellis representation, a distinct reduction in complexity compared with a general representation takes place with the result that decoding of the code word (generation of the soft outputs at the positions of the code word) also becomes possible in real time.

[0011] A further development consists in that the decoding rule for each position of the code word is determined by

$$L(U_i|y) = \ln \left(\frac{\sum\limits_{c \in \Gamma^i(+1)} \exp\left(-\frac{(y-c)^T(y-c)}{2\sigma^2}\right)}{\sum\limits_{c \in \Gamma^i(-1)} \exp\left(-\frac{(y-c)^T(y-c)}{2\sigma^2}\right)} \right), \quad \text{for } i = 1, \ldots, K, \qquad (1)$$

[0012]

[0013] where

[0014] $L(U_i|y)$ is a safety measure (soft output) for the i-th position of the code word to be determined;

[0015] y is a demodulation result to be decoded;

[0016] c is a code word;

[0017] $\Gamma^i(\pm 1)$ are all code words for $u_i = \pm 1$;**and**

[0018] σ^2 is a variance (channel disturbance).

[0019] Another further development [~~consists in~~]**provides** that the equation (1) is solved by utilizing a characteristic of a convolutional code used in the coding (and correspondingly in the decoding) which determines states in accordance with a shift register operation used during the convolution, from which states, in turn, the trellis representation is obtained.

[0020] In an additional further development, the trellis representation is run through in a predetermined direction in order to recursively calculate terms A_m and A_m respectively. Into this calculation rule, node weights $\mu_m(s)$ which are determined by the demodulation result y enter at the nodes (s, m) of the trellis representation. The terms A_m and A_m are described by

[0021]

$$\tilde{A}_m(E) = \sum_{s \in E} A_m(s), \quad \text{for } m \in \mathbb{N} \tag{2}$$

[0022] with

[0023]

$$A_m(s) = \mu_m(s) \sum_{t \in W(s, V_m)} A_{m-1}(t), \quad \text{for } m \in \mathbb{N} \tag{3}$$

[0024] and a starting value

[0025]

$$A_0(s) = \begin{cases} 1 & : \text{ for } s = s_0, \\ 0 & : \text{ else} \end{cases} \tag{4}$$

[0026] A more detailed discussion of the forms of description listed here can also be found in the description of the exemplary embodiment.

[0027] One embodiment [~~consists in~~]**provides** that mappings B_m are determined by [~~means~~]**way** of the trellis representation, the trellis representation being processed in opposition to the predetermined direction. The term B_m is determined by

$$B_m(s) = \mu_{Q-m+1}(s) \sum_{t \in T(s, V_{Q-m+2})} B_{m-1}(t), \qquad \text{for } 1 \leq m \leq Q, \qquad (5)$$

[0028]

[0029] where

$$B_0(s) = \begin{cases} 1 & : \text{ for } s = s_0, \\ 0 & : \text{ else} \end{cases} \qquad (6)$$

[0030]

[0031] is determined for terminating the recursion.

[0032] Furthermore, terms A^i_α can be determined by again running through the trellis representation taking into consideration the terms A_m and B_m already determined. In particular, the terms [~~are determined in accordance with~~] A^i_α **are determined in accordance with**

$$A^i_\alpha(y) = \sum_{s \in S} A_{j-1}(s) \sum_{t \in T(s, V^i_j(\alpha))} B_{Q-j+1}(t). \qquad (7)$$

[0033]

[0034] In a further embodiment, the K positions of the decoded code word are determined in accordance with

$$L(U_i|y) = \ln\left(\frac{A^i_{+1}(y)}{A^i_{-1}(y)}\right), \qquad i = 1, \ldots, K. \qquad (8)$$

[0035]

[0036] In particular, an AWGN (Additive Gaussian White Noise) channel model is used for the derivation. The method presented can also be used for other channel models, especially for channel models used in mobile radio.

[0037] Another embodiment relates to the use of the method in a mobile radio network, especially the GSM network.

[0038] It is also a further development that, after the soft outputs have been determined, there is a "hard" correlation of the analogue values with the Boolean values ±1. In this arrangement, the nearest Boolean value is in each case determined for correlating the analogue value.

[0039] The soft output values determined can be used as input values for further decoding when concatenated codes are used.

[0040] To achieve the object, an arrangement for decoding a predetermined code word is also specified in which a processor unit is provided which is set up in such a manner that

[0041] 1. the code word comprises a number of positions having different values;

[0042] 2. each position of the code word can be correlated with a soft output value by performing the correlation on the basis of a trellis representation;**and**

[0043] 3. the decoding of the code word can be determined by the correlation of the individual positions of the code word.

[0044] This arrangement is particularly suitable for performing the method according to the invention or one of its further developments explained above.

BRIEF DESCRIPTION OF THE DRAWINGS

[0045] In the text which follows, exemplary embodiments of the invention will be shown and explained with reference to the [~~drawing~~]**drawings**, in which:

[0046] Figure 1 [~~shows~~]**is a block diagram showing** a representation of digital information transmission;

[0047] Figure 2 [~~shows~~]**is** an algorithm in pseudocode notation for progressing in the trellis diagram observing all states for the calculation of node weights;

[0048] Figure 3 [~~shows~~]**is** an algorithm in pseudocode notation for determining soft outputs (general case);

[0049] Figure 4 [~~shows~~]**is** an algorithm in pseudocode notation for determining soft outputs (special case: binary state transition);**and**

[0050] Figure 5 [~~shows~~]**is a block diagram of** a processor unit.

DETAILED DESCRIPTION OF THE INVENTION

[0051] [~~In the~~]**The** text which follows**describes in greater detail**, first the convolutional code, then the reduction in complexity in the calculation of soft outputs and,

finally, an algorithmic translation of the reduction in complexity[~~ are described in greater detail~~].

TERMINATED CONVOLUTIONAL CODE

[0052] In communication technology, terminated convolutional codes are mostly used in concatenation with other systematic or unsystematic block codes. In particular, the decoding result of a convolutional decoder is used as the input for another decoder.

[0053] To ensure the lowest possible error rate, it is necessary to supply "soft" decoding decisions instead of "hard" ones in the convolutional decoding for the further decoder, i.e.**,** to generate a tuple of "soft" values (soft outputs) from R instead of a tuple of "hard" Boolean (±1) values. The absolute value of the respective "soft" decision then provides a safety measure for the correctness of the decision.

[0054] In principle, these soft outputs can be calculated in accordance with equation (1), depending on the channel model. However, the numeric complexity for calculating a soft output is $O(2^K)$, where K specifies the number of information bits. If K is realistically large, **then** these formulae can [~~thus~~]not be evaluated, in particular, since such a code word must be calculated again every few milliseconds (**a** real-time requirement).

[0055] One consequence [~~was~~]**of this is** that soft outputs [~~were~~]**are** dispensed with (with all consequences for the word and bit error rates) or, respectively, fewer elaborate approximations [~~were~~]**are** performed for determining the soft outputs.

[0056] In the text which follows, a possibility for terminated convolutional codes is specified with the aid of which this complexity can be reduced to $O(K)$ in a trellis representation for calculating all soft outputs, i.e.[~~ it~~]**, this solution** provides the possibility for a precise evaluation of equation (1).

[0057] In the text which follows, the bits of the code are represented in $\{\pm 1\}$ representation. In comparison with a $\{0, 1\}$ representation, which is often used in information technology, -1 corresponds to 1 and 1 corresponds to 0.

[0058] On a body $\{\pm 1\}$, addition $\oplus$ and multiplication $\odot$ are defined as follows:

$$-1 \oplus -1 = 1 \qquad -1 \odot -1 = -1$$

$$-1 \oplus 1 = -1 \qquad -1 \odot 1 = 1$$

$1 \oplus -1 = -1$ $\qquad$ $1 \odot -1 = 1$

$1 \oplus 1 = 1$ $\qquad$ $1 \odot 1 = 1$

[0059] The coding is done with the aid of a "shift register" into which bit blocks (input blocks) of the information bits are written with each clock pulse. The combination of the bits of the shift register then generates one bit block of the code word. The shift register is pre-assigned +1 bits in each case. To terminate the coding (termination) blocks of tail zeros (+1) are shifted in afterwards. As has been mentioned initially, check bits by [~~means~~]**way** of which bit errors can be corrected are correlated with the information bits by [~~means~~]**way** of coding.

[0060] The following are defined for the further embodiments:

[0061] $b \in \mathrm{N}$ number of input bits per block**;**

[0062] $V{:=}\{\pm 1\}^b$ set of state transition signs**;**

[0063] $a \in \mathrm{N}$ number of input blocks**;**

[0064] $K{:=}\, a \cdot b$ number of information bits without tail zeros **;**

[0065] $k \in \mathrm{N}, k \geq 2$ block length of the shift register, penetration depth**;**

[0066] $L{:=}\, k \cdot b$ bit length of the shift register**;**

[0067] $S{:=}\{\pm 1\}^L$ set of shift register signs**;**

[0068] $n \in \mathrm{N}$ number of output bits per block**;**

[0069] $Q{:=}\, a + k - 1$ number of state transitions, input blocks + zeros**;**

[0070] $N{:=}\, n \cdot Q$ number of code bits**; and**

[0071] $R{:=}\frac{b}{n}$ code rate**.**

[0072] It should be noted here that the code rate is not K/N since the information bits have been counted without the zeros (+1) of the convolutional termination.

[0073] Furthermore, $s_0 \in S$ and $v_0 \in V$ are assumed to be the respective zero elements, i.e.**,**

[0074]
$$s_0 = (+1, \ldots, +1)^{\mathsf{T}}, \quad v_0 = (+1, \ldots, +1)^{\mathsf{T}}. \tag{9}$$

[0075] The state transition function of the shift register is assumed to be

$$T: \; S \times V \to S, \tag{10}$$

$$(s, v) \mapsto (s^{b+1}, \ldots, s^{L}, v^{1}, \ldots, v^{b})^{\mathsf{T}}. \tag{11}$$

[0076]

[0077] The terminated convolutional code is defined by the characterizing subsets

$$M_1, \ldots, M_n \subseteq \{1, \ldots, L\}, \tag{12}$$

[0078]

[0079] (combination of register bits, alternatively in polynomial representation).

[0080] The current register content is coded via

$$C: \; S \to \{\pm 1\}^{n}, \tag{13}$$

$$s \mapsto C(s) \text{ where } C_j(s) := \bigoplus_{i \in M_j} s^{i}, \quad \text{for} \quad 1 \leq j \leq n. \tag{14}$$

[0081]

[0082] where s^i is the i-th component of s.

[0083] Finally, the coding of an information word is defined by [~~means~~]**way** of

$$\varphi: \; \{\pm 1\}^{K} \to \{\pm 1\}^{N}, \tag{15}$$

$$u \mapsto \begin{pmatrix} C(s_1) \\ \vdots \\ (s_Q) \end{pmatrix}, \tag{16}$$

[0084]

where $s_0 \in S$ is the zero state (zero element),

$$u = \begin{pmatrix} \nu_1 \\ \vdots \\ \nu_a \end{pmatrix}, \quad \nu_i \in V, \quad 1 \leq i \leq a, \tag{17}$$

$$\nu_i := v_0, \quad a+1 \leq i \leq Q, \tag{18}$$

[0085]

[0086] and furthermore

[0087]

$$s_i := T(s_{i-1}, \nu_i), \quad 1 \leq i \leq Q. \tag{19}$$

[0088] According to the definition of T, the following is obtained

[0089]

$$s_{Q+1} := T(s_Q, v_0) = s_0. \tag{20}$$

[0090] Accordingly, the set of all code words is

[0091]

$$\varphi\left(\{\pm 1\}^K\right) := \left\{\varphi(u) \in \{\pm 1\}^N;\ u \in \{\pm 1\}^K\right\}. \tag{21}$$

[0092] Often, polynomials

[0093]

$$p_j \in \{0,1\}[D] \quad \text{where} \quad \deg(p_j) \leq L-1$$

[0094] are used instead of the sets M_j for code definition, i.e.,

[0095]

$$p_j(D) = \sum_{i=0}^{L-1} \gamma_{i,j} D^i, \tag{22}$$

[0096] with

[0097]

$$\gamma_{i,j} \in \{0,1\} \qquad \begin{array}{l} i = 0, \ldots, L-1, \\ j = 1, \ldots, n. \end{array}$$

[0098] The following transformations then apply for $j = 1, \ldots, n$:

[0099]

$$M_j = \{i \in \{1, \ldots, L\};\ \gamma_{L-i,j} = 1\} \tag{23}$$

$$p_j(D) = \sum_{i \in M_j} D^{L-i}. \tag{24}$$

BLOCK CODE REPRESENTATION

[00100] Since a terminated convolutional code is a block code, the code bits c_j, $1 \le j \le N$ can also be represented from the information bits u_i, $1 \le i \le K$, with index sets J_j, as follows:

$$c_j := \bigoplus_{i \in J_j} u_i, \quad \text{for } 1 \le j \le N, \tag{25}$$

[00101]

[00102] where

$$J_1, \ldots, J_N \subseteq \{1, \ldots, K\}. \tag{26}$$

[00103]

[00104] The index sets J_j can be calculated directly from the above index sets M_m of the code definition.

[00105] Consider

$$j = n(q-1) + m, \quad q = 1, \ldots, Q, \; m = 1, \ldots, n. \tag{27}$$

$$c_j = C_m(s_q) = \bigoplus_{i \in M_m} (s_q)^i = \bigoplus_{i \in M_m} u_{i+b(q-k)}, \tag{28}$$

[00106]

[00107] where $u_i := +1$ for $i \notin \{1,\ldots,K\}$.

[00108] Furthermore,

$$c_j = \bigoplus_{i-b(q-k) \in M_m} u_i = \bigoplus_{i \in M_m + b(q-k)} u_i, \tag{29}$$

[00109]

[00110] and it thus follows for $j = 1,\ldots,N$ that

$$\begin{aligned} J_j &= \{1, \ldots, K\} \cap (M_m + b(q-k)) \\ &= \{i \in \{1, \ldots, K\};\ i - b(q-k) \in M_m\}. \end{aligned} \tag{30}$$

[00111]

EXAMPLE: SACCH CONVOLUTIONAL CODE

[00112] In the above terminology, the convolutional code described in section 4.1.3 of the GSM Technical Specification GSM 05.03, Version 5.2.0 (channel coding) is:

[00113] $b = 1$ number of input bits per block__;__

[00114] $V=\{\pm 1\}$ set of state transition signs__;__

[00115] $a = 224$ number of input blocks__;__

[00116] $K = 224$ number of information bits without tail zeros__;__

[00117] $k = 5$ block length of the shift register, depth of penetration__;__

[00118] $L = 5$ bit length of the shift register__;__

[00119] $S = \{\pm 1\}^5$ set of shift register signs__;__

[00120] $n = 2$ number of output bits per block__;__

[00121] $Q = 228$ number of state transitions, input blocks + zeros__;__

[00122] $N = 456$ number of code bits__;__

[00123] $R = \frac{1}{2}$ code rate__;__

[00124] $M_1 = \{1,2,5\}$ characterizing set; polynomial: $1+D^3+D^4$__; and__

[00125] $M_2 = \{1,2,4,5\}$ characterizing set; polynomial: $1+D+D^3+D^{[4]4}$__.__

Soft Outputs in an AWGN Channel Model

[00126] In the text which follows, calculation rules for determining the soft outputs are derived, especially for the sake of clarity.

[00127] For this purpose, a probability space (Ω, S, P) and a K-dimensional random variable $U:\Omega\rightarrow\{\pm 1\}^K$ are considered which have the properties

- The components $U_1,\ldots,U_K : \Omega\rightarrow\{\pm 1\}$ are stochastically independent.
- The following holds for $i = 1,\ldots,K$

$$P(\{\omega \in \Omega;\ U_i(\omega) = -1\}) = P(\{\omega \in \Omega;\ U_i(\omega) = +1\}). \qquad (31)$$

[00128]

[00129] Figure 1 shows a representation of digital telecommunication. A unit consisting of source 201, source encoder 202 and crypto-encoder 203 determines an information item $u \in \{\pm 1\}^K$ which is used as input for one (or possibly more) channel encoder(s) 204. The channel encoder 204 generates a code word $c \in \{\pm 1\}^N$ which is fed into

a modulator 205 and is transmitted via a disturbed physical channel 206 to a receiver where it is determined to become a real-value code word $y \in \mathrm{R}^N$ in a demodulator 207. This code word is converted into a real-value information item in a channel decoder 208. If necessary, a "hard" correlation with the Boolean values ±1 can also be made in a further decoder so that the received information is present in Boolean notation. The receiver is completed by a unit of crypto-decoder 209, source decoder 210 and sink 211. The two crypto-encoder 203 and crypto-decoder 209 units are optional in this arrangement.

[00130] The information to be reconstructed, $u \in \{\pm 1\}^K$, of the crypto-encoder 203 is interpreted as implementation of the random variables U since nothing is known about the choice of u in the receiver.

[00131] Thus, the output $c \in \{\pm 1\}^N$ of the channel encoder 204 is an implementation of the random variables $\varphi(U)$.

[00132] The output $y \in \mathrm{R}^N$ of the demodulator 207 is interpreted as implementation of the random variables

$$Y : \Omega \to \mathbb{R}^N, \tag{32}$$

$$\omega \mapsto \varphi(U(\omega)) + Z(\omega), \tag{33}$$

[00133]

[00134] a random variable $Z : \Omega \to \mathrm{R}^N$ representing the channel disturbances in the physical channel 206.

[00135] In the text which follows, an AWGN channel model is assumed, i.e., Z is a $N(0,\sigma^2 I_N)$ normally distributed random variable which is stochastically independent of U and, respectively, $\varphi(U)$. The variance σ^2 is calculated from the ratio between noise power density and mean energy in the channel 206 and is here assumed to be known.

[00136] The unknown output $u \in \{\pm 1\}^K$ of the crypto-encoder is to be reconstructed on the basis of an implementation y of Y. To estimate the unknown quantities $u_1,\dots,u_K$, the distribution of the random variables U is investigated given the condition that y has been received.

[00137] The consequence of the fact that the random variable Y is a steady random variable is that the consideration of U under the condition that y has been received ($Y(\hat{\omega})=y$) is extremely complicated.

[00138] Firstly, the following is defined for $i \in \{1,...,K\}$ and $\alpha \in \{\pm 1\}$

[00139]
$$\Gamma^i(\alpha) := \left\{\varphi(u);\ u \in \{\pm 1\}^K;\ u_i = \alpha\right\}. \tag{34}$$

[00140] In a preparatory step, the following quantities are considered for $\epsilon > 0$, paying attention to the injectivity of the coding map φ:

[00141]
$$\begin{aligned} L_\epsilon(U_i|y) &:= \ln\left(\frac{P(\{\omega \in \Omega;\ U_i(\omega) = +1\} \mid \{\omega \in \Omega;\ Y(\omega) \in M_{y,\epsilon}\})}{P(\{\omega \in \Omega;\ U_i(\omega) = -1\} \mid \{\omega \in \Omega;\ Y(\omega) \in M_{y,\epsilon}\})}\right) \\ &= \ln\left(\frac{\sum\limits_{c \in \Gamma^i(+1)} P(\{\omega \in \Omega;\ \varphi(U(\omega)) = c\} \mid \{\omega \in \Omega;\ Y(\omega) \in M_{y,\epsilon}\})}{\sum\limits_{c \in \Gamma^i(-1)} P(\{\omega \in \Omega;\ \varphi(U(\omega)) = c\} \mid \{\omega \in \Omega;\ Y(\omega) \in M_{y,\epsilon}\})}\right), \end{aligned} \tag{35}$$

[00142] for $i=1,...,K$, where $M_{y,\varepsilon} := [y_1, y_1+\varepsilon] \times ... \times [y_N, y_N+\varepsilon]$.

[00143] Using the theorem by Bayes, the following is obtained:

[00144]
$$\begin{aligned} L_\epsilon(U_i|y) &= \ln\left(\frac{\sum\limits_{c \in \Gamma^i(+1)} P(\{\omega \in \Omega;\ Y(\omega) \in M_{y,\epsilon}\} \mid \{\omega \in \Omega;\ \varphi(U(\omega)) = c\})}{\sum\limits_{c \in \Gamma^i(-1)} P(\{\omega \in \Omega;\ Y(\omega) \in M_{y,\epsilon}\} \mid \{\omega \in \Omega;\ \varphi(U(\omega)) = c\})}\right) \\ &= \ln\left(\frac{\sum\limits_{c \in \Gamma^i(+1)} \int\limits_{M_{y,\epsilon}} \exp\left(-\frac{(x-c)^\top (x-c)}{2\sigma^2}\right) dx}{\sum\limits_{c \in \Gamma^i(-1)} \int\limits_{M_{y,\epsilon}} \exp\left(-\frac{(x-c)^\top (x-c)}{2\sigma^2}\right) dx}\right). \end{aligned} \tag{36}$$

[00145] Considering then the limiting process of $L_\varepsilon(U_i|y)$ for $\varepsilon \downarrow 0$ by using L'Hospital's rule several times, the soft output $L(U_i|y)$ is obtained for each symbol as in equation (1).

[00146] Since

[00147]
$$\Gamma^i(+1) \cup \Gamma^i(-1) = \{\pm 1\}^K$$

[00148] holds true, a total of $O(2^K)$ numeric operations are necessary for evaluating equation (1).

[00149] The vector $L(U_\bullet|y) \in \mathbb{R}^K$ is the result of decoder 208.

Reduction of Complexity in the Determination of the Soft Outputs

Soft-Output Determination for Convolutional Codes

[00150] Firstly, the special characteristics of the terminated convolutional coding are used for providing an organized representation of the soft-output formula (1).

[00151] For an arbitrary, but preselected output $y \in \mathrm{R}^N$ of the demodulator 207, the following weighting function (a Viterbi metric) of code words is considered:

$$F : \{\pm 1\}^N \to \mathbb{R}_0^+, \tag{37}$$

$$c \mapsto \sum_{j=1}^{N} (y_j - c_j)^2. \tag{38}$$

[00152]

[00153] For permissible code words $c \in \{\pm 1\}^N$, i.e. $c \in \varphi(\{\pm 1\}^K)$, $F(c)$ can be reduced as follows, using the shift register representation:

$$F(c) = \sum_{q=1}^{Q} \underbrace{\sum_{j=1}^{n} \left(y_{n(q-1)+j} - C_j(\tilde{s}_q^c)\right)^2}_{=:\Delta F_q(\tilde{s}_q^c)}, \tag{39}$$

[00154]

[00155] where $\tilde{s}_q^c$ stands for the q-th state of the shift register in the (unambiguous) generation of the word c.

[00156] Then the following is defined for I=1,...,K and $\alpha \in \{\pm 1\}$:

$$A_\alpha^i(y) := \sum_{c \in \Gamma^i(\alpha)} \exp\left(-\frac{(y-c)^\top (y-c)}{2\sigma^2}\right) = \sum_{c \in \Gamma^i(\alpha)} \prod_{q=1}^{Q} \exp\left(-\frac{1}{2\sigma^2} \Delta F_q(\tilde{s}_q^c)\right). \tag{40}$$

[00157]

[00158] Thus, the following holds true for the soft outputs

$$L(U_i|y) = \ln\left(\frac{A_{+1}^i(y)}{A_{-1}^i(y)}\right), \quad i = 1, \ldots, K. \tag{41}$$

[00159]

[00160] In the text which follows, the values $A_\alpha^i(y)$ are determined with the aid of a trellis diagram /representation[~~(also: trellis diagram or trellis representation)~~].

[00161] To reduce the complexity of calculation, the following procedure is adopted in the following sections:

[00162] • generalization of [~~by mappings~~][~~,~~] A_{α}^{i} **by mappings** $\widetilde{A}_m$**:**

[00163] • Recursive representation of $\widetilde{A}_m$ by mappings A_m, the values of which are calculated with a "from left to right" run through a trellis diagram.

[00164] • Reversal of the recursion by mappings B_m, the values of which are calculated with a "from right to left" run through a trellis diagram.

[00165] • joint calculation of all A_{α}^{i} by [~~means~~]**way** of a further run through a trellis diagram by using A_m and B_m.

[00166] The trellis diagram is here a set

$$\mathcal{T} = \{(s,q);\ s \in S,\ q = 0,\ldots,Q+1\} \tag{42}$$

[00167]

[00168] The elements (s,q) of this set are also called the nodes in the trellis diagram, s representing a state and q being considered as a dynamic value (especially time).

GENERAL RECURSIVE REPRESENTATION

[00169] **Firstly, some definitions are needed for representing the** [~~Firstly, some definitions are needed for representing the~~] A_{α}^{i} in a generalized form which allows later transformation. For this reason, the following is determined

$$s_1^u := T(s_0, u_1), \quad u \in V^m = V \times \ldots \times V,\ m \geq 1, \tag{43}$$
$$s_j^u := T(s_{j-1}^u, u_j) \quad u \in V^m,\ m \geq j > 2, \tag{44}$$

[00170]

[00171] i.e., s_j^u represents the state of the shift register after j shifts of the register with the input symbols $u_1,\ldots,u_j$.

[00172] Furthermore, sets $V_j \subseteq V, j \in \mathbb{N}$, which contain the permissible state transition symbols in the j-th step, are considered. Furthermore, product sets are defined as

$$U_m := V_1 \times \ldots \times V_m \subseteq V^m, \quad m \in \mathbb{N}, \tag{45}$$

[00173]

[00174] i.e., U_m contains the first m components of the permissible input words.

[00175] For $q \in$ N, mappings

$$\mu_q : S \to \mathbb{R} \qquad (46)$$

[00176]

[00177] are considered and for $m \in$ N and input word sets $U_m \subseteq V^m$, mappings are defined as follows

$$\tilde{A}_m : \wp(S) \to \mathbb{R}, \qquad (47)$$

$$E \mapsto \sum_{\substack{(u \in U_m) \\ \wedge (s_m^u \in E)}} \prod_{j=1}^{m} \mu_j(s_j^u), \qquad (48)$$

[00178]

[00179] i.e., summing over all permissible input words, the shift register of which reaches a final state in E, is performed. If there are no such input words, the sum is determined as 0 over an empty index set.

[00180] In addition, a mapping is determined as

$$W : S \times \wp(V) \to \wp(S), \qquad (49)$$

$$(t, \hat{V}) \mapsto \left\{ s \in S;\ \exists \hat{v} \in \hat{V} \ni T(s, \hat{v}) = t \right\}, \qquad (50)$$

[00181]

[00182] i.e., W maps $(t, \hat{V})$ into the sets of all states which can reach the state t with a transition symbol from [-] $\hat{V}$.

[00183] The following holds true for $m \geq 2$, $E \subseteq S$

$$
\begin{aligned}
\tilde{A}_m(E) &= \sum_{\substack{(u \in U_m) \\ \wedge (s_m^u \in E)}} \prod_{j=1}^{m} \mu_j(s_j^u) \\
&= \sum_{s \in E} \sum_{\substack{(u \in U_m) \\ \wedge (s_m^u = s)}} \prod_{j=1}^{m} \mu_j(s_j^u) \\
&= \sum_{s \in E} \mu_m(s) \sum_{\substack{(u \in U_m) \\ \wedge (s_m^u = s)}} \prod_{j=1}^{m-1} \mu_j(s_j^u) \\
&= \sum_{s \in E} \mu_m(s) \sum_{\substack{(u \in U_{m-1}) \\ \wedge (s_{m-1}^u \in W(s, V_m))}} \prod_{j=1}^{m-1} \mu_j(s_j^u) \\
&= \sum_{s \in E} \mu_m(s) \tilde{A}_{m-1}\left(W(s, V_m)\right). \qquad (51)
\end{aligned}
$$

[00184]

[00185] In the transformation in the last step but one, attention must be paid to the fact that there is **exactly one** transition symbol $v \in V_m$ with $T\left(s_{m-1}^{u}, v\right)=s$, if [~~is in *W(s,V_m)*, i.e.~~][~~it is not necessary to take account of any multiplicities.~~] s_{m-1}^{u} **is in *W(s,V_m)*, i.e.,** ~~is in *W(s,V_m)*, i.e.~~ **it is not necessary to take account of any multiplicities.**

[00186] Consider, then, the following for $m \geq 2$ mappings

$$
\begin{aligned}
A_m : S &\to \mathbb{R}, && (52) \\
s &\mapsto \mu_m(s)\tilde{A}_{m-1}\left(W(s, V_m)\right). && (53)
\end{aligned}
$$

[00187]

[00188] Thus, a recursion formula can be derived for $m \geq 3$:

$$
\begin{aligned}
A_m(s) &= \mu_m(s)\tilde{A}_{m-1}\left(W(s, V_m)\right) \\
&= \mu_m(s) \sum_{t \in W(s, V_m)} \mu_{m-1}(t)\tilde{A}_{m-2}\left(W(t, V_{m-1})\right) \\
&= \mu_m(s) \sum_{t \in W(s, V_m)} A_{m-1}(t). \qquad (54)
\end{aligned}
$$

[00189]

[00190] Furthermore:

$$
\begin{aligned}
A_2(s) &= \mu_2(s)\tilde{A}_1\left(W(s, V_2)\right) \\
&= \mu_2(s) \sum_{\substack{(u \in U_1) \\ \wedge (s_1^u \in W(s, V_2))}} \mu_1(s_1^u) \\
&= \mu_2(s) \sum_{t \in W(s, V_2)} \mu_1(t)\delta_{s_0 \in W(t, V_1)} \\
&= \mu_2(s) \sum_{t \in W(s, V_2)} \underbrace{\mu_1(t) \sum_{\hat{t} \in W(t, V_1)} \underbrace{\delta_{\hat{t}=s_0}}_{=:A_0(\hat{t})}}_{=:A_1(t)}
\end{aligned} \tag{55}
$$

[00191]

[00192] In summary, the following thus holds true for $s \in S$, $E \subseteq S$:

$$
A_0(s) = \begin{cases} 1, & \text{for} \quad s = s_0, \\ 0, & \text{otherwise} \end{cases}, \tag{56}
$$

$$
A_m(s) = \mu_m(s) \sum_{t \in W(s, V_m)} A_{m-1}(t), \quad \text{for} \quad m \in \mathbb{N}, \tag{57}
$$

$$
\tilde{A}_m(E) = \sum_{s \in E} A_m(s), \quad \text{for} \quad m \in \mathbb{N}. \tag{58}
$$

[00193]

[00194] The sets $W(s, V_m)$ can be represented constructively. For this purpose, two further mappings are considered. The following is defined

$$
\tau : S \to V, \tag{59}
$$

$$
s = (s^1, \ldots, s^L)^\top \mapsto (s^{L-b+1}, \ldots, s^L)^\top, \tag{60}
$$

[00195]

[00196] i.e., if the state s is the result of a state transition, then $\tau(s)$ was the associated state transition symbol.

[00197] Furthermore

$$
\hat{T} : V \times S \to S, \tag{61}
$$

$$
(v, s) \mapsto (v^1, \ldots, v^b, s^1, \ldots, s^{L-b})^\top, \tag{62}
$$

[00198]

[00199] is defined, i.e., $\hat{T}$ reverses the direction of the shift register operation.

[00200] The following then holds

$$T\left(\hat{T}(v,s),\tau(s)\right) = s, \quad \text{for all} \quad s \in S,\ v \in V \tag{63}$$

[00201]

[00202] and for all $t \in S$ and $\hat{V} \subseteq V$, it also holds true that

$$\begin{aligned} W(t,\hat{V}) &= \left\{s \in S;\ \exists \hat{v} \in \hat{V} \ni T(s,\hat{v}) = t\right\} \\ &= \begin{cases} \left\{\hat{T}(v,t);\ v \in V\right\}, & \text{if} \quad \tau(t) \in \hat{V}, \\ \emptyset, & \text{else.} \end{cases} \end{aligned} \tag{64}$$

[00203]

[00204] Thus, the recursion formula (57) for $A_m(s)$ can be written down constructively as follows:

$$\begin{aligned} A_m(s) &= \mu_m(s) \sum_{t \in W(s,V_m)} A_{m-1}(t) \\ &= \begin{cases} \mu_m(s) \sum_{v \in V} A_{m-1}\left(\hat{T}(v,s)\right), & \text{if} \quad \tau(s) \in V_m, \\ 0, & \text{else.} \end{cases} \end{aligned} \tag{65}$$

[00205]

[00206] It should be noted that in this section, no restrictions were set for the set V of the state transition symbols and for the sets $V_j \in \wp(V)$.

REVERSAL OF RECURSION

[00207] In the text which follows, a recursion in the "reverse direction" compared with the above recursion is described. This new recursion is defined with the aid of the recursion formula (57) for $A_m(s)$.

[00208] The following is assumed for this purpose

$$T(t,\hat{V}) := \left\{T(t,\hat{v});\ \hat{v} \in \hat{V}\right\}, \quad \text{for} \quad t \in S,\ \hat{V} \subseteq V \tag{66}$$

[00209]

[00210] and for $M \in$ N, $0 \leq m \leq Q$, the following mappings are considered

$$B_m : S \rightarrow \mathbb{R}, \tag{67}$$

[00211]

[00212] with the following recursive characteristic:

$$\begin{aligned}
&\sum_{s\in S} A_m(s) \sum_{t\in T(s,V_{m+1})} B_{Q-m}(t) = \\
&= \sum_{s\in S} \mu_m(s) \sum_{\hat{t}\in W(s,V_m)} A_{m-1}(\hat{t}) \sum_{t\in T(s,V_{m+1})} B_{Q-m}(t) \\
&= \sum_{\hat{t}\in S} \sum_{s\in T(\hat{t},V_m)} \mu_m(s) A_{m-1}(\hat{t}) \sum_{t\in T(s,V_{m+1})} B_{Q-m}(t) \\
&= \sum_{\hat{t}\in S} A_{m-1}(\hat{t}) \sum_{s\in T(\hat{t},V_m)} \underbrace{\mu_m(s) \sum_{t\in T(s,V_{m+1})} B_{Q-m}(t)}_{=:B_{Q-m+1}(s)},
\end{aligned}$$

[00213]

[00214] i.e.,

$$\sum_{s\in S} A_m(s) \sum_{t\in T(s,V_{m+1})} B_{Q-m}(t) = \sum_{s\in S} A_{m-1}(s) \sum_{t\in T(s,V_m)} B_{Q-m+1}(t). \tag{68}$$

[00215]

[00216] By applying equation (68) several times, the following is obtained for an arbitrary $j \in \{1,\ldots,m+1\}$

$$\sum_{s\in S} A_m(s) \sum_{t\in T(s,V_{m+1})} B_{Q-m}(t) = \sum_{s\in S} A_{j-1}(s) \sum_{t\in T(s,V_j)} B_{Q-j+1}(t). \tag{69}$$

[00217]

[00218] According to the above definition, the recursion formula is thus

$$B_m(s) = \mu_{Q-m+1}(s) \sum_{t\in T(s,V_{Q-m+2})} B_{m-1}(t), \qquad \text{for } 1 \le m \le Q. \tag{70}$$

[00219]

[00220] To terminate the recursion, the following are defined

$$B_0(s) = \begin{cases} 1, & \text{for } s = s_0, \\ 0, & \text{else} \end{cases}. \tag{71}$$

[00221]

[00222] Given this termination and the equations (58) and (69),

$$\tilde{A}_Q(W(s_0, V_{Q+1}))$$

[00223]

[00224] can be represented for $V_{Q+1} := \{v_0\}$ and with an arbitrary $j \in \{1,...,Q+1\}$ as follows:

$$\begin{aligned}\tilde{A}_Q(W(s_0, V_{Q+1})) &= \sum_{s \in W(s_0, V_{Q+1})} A_Q(s) \\ &= \sum_{s \in S} A_Q(s) \sum_{t \in T(s,\{v_0\})} B_0(t) \\ &= \sum_{s \in S} A_Q(s) \sum_{t \in T(s,V_{Q+1})} B_0(t) \\ &= \sum_{s \in S} A_{j-1}(s) \sum_{t \in T(s,V_j)} B_{Q-j+1}(t). \qquad (72)\end{aligned}$$

[00225]

[00226] Note: in the evaluation of (72), V_j is not included in the calculation of the A_m and B_m needed.

CALCULATION OF A_α^i

[00227] Using the preliminary work from the preceding sections, [~~can now be calculated in a simple manner.~~] A_α^i can now be calculated in a simple manner.

[00228] For this purpose, the following are defined:

$$V_j := V, \quad \text{for} \quad j \in \{1, \ldots, a\}, \qquad (73)$$
$$V_j := \{v_0\}, \quad \text{for} \quad j \in \{a+1, \ldots, Q+1\}, \qquad (74)$$

[00229]

[00230] i.e., all permissible code words are defined via the states s_J^u with

$$u \in U_Q = V_1 \times \ldots \times V_Q$$

[00231]

[00232] The code words used in the calculation of the A_α^i are restricted by $u_i=\alpha$. For an arbitrary but fixed choice [~~of i~~]**of i** $\in$[~~$\{1,\ldots,K\}$, there is exactly one j~~] **$\{1,\ldots,K\}$, there is exactly one j** $\in$[~~$\{1,\ldots,a\}$ and exactly one~~] **$\{1,\ldots,a\}$ and exactly one** $\hat{\imath} \in \{1,\ldots,b\}$ with

$$i = (j-1)\cdot b + \hat{\imath}. \tag{75}$$

[00233]

[00234] Furthermore, the following are defined for an arbitrary but fixed choice of $\alpha \in \{\pm 1\}$:

$$V_j^i(\alpha) := \{v \in V;\ v_{\hat{\imath}} = \alpha\} \tag{76}$$
$$U_Q^i(\alpha) := V_1 \times \ldots \times V_{j-1} \times V_j^i(\alpha) \times V_{j+1} \times \ldots \times V_Q \subset U_Q, \tag{77}$$

[00235]

i.e.**,** the code words from $\Gamma^i(\alpha)$ are determined via the states A_α^i with $u \in$ [~~.~~]$U_Q^i(\alpha)$**.**

[00236] For an arbitrary, but fixed choice of $y \in \mathrm{R}^N$, define for $q \in \{1,\ldots,Q\}$

$$\mu_q : S \to \mathbb{R}, \tag{78}$$
$$s \mapsto \exp\left(-\frac{1}{2\sigma^2}\sum_{j=1}^{n}\left(y_{n(q-1)+j} - C_j(s)\right)^2\right) = \exp\left(-\frac{1}{2\sigma^2}\Delta F_q(s)\right). \tag{79}$$

[00237]

[00238] According to the definition of the convolutional code, the following holds true for all s_Q^u with $u \in U_Q$:

$$s_{Q+1}^u = T(s_Q^u, u_{Q+1}) = s_0, \quad u_{Q+1} \in V_{Q+1} = \{v_0\}, \tag{80}$$

[00239]

[00240] i.e.**,**

$$s_Q^u \in W(s_0, V_{Q+1}). \tag{81}$$

[00241]

[00242] Taking account of equation (72), the following thus holds true:

$$
\begin{aligned}
A_\alpha^i(y) &= \sum_{c\in\Gamma^i(\alpha)} \prod_{q=1}^{Q} \exp\left(-\frac{1}{2\sigma^2}\Delta F_q(\tilde{s}_q^c)\right) \\
&= \sum_{u\in U_Q^i(\alpha)} \prod_{q=1}^{Q} \mu_q(s_q^u) \\
&= \sum_{\substack{(u\in U_Q^i(\alpha)) \\ \wedge(s_Q^u\in W(s_0,V_{Q+1}))}} \prod_{q=1}^{Q} \mu_q(s_q^u) \\
&= \tilde{A}_Q(W(s_0,V_{Q+1})) \\
&= \sum_{s\in S} A_{j-1}(s) \sum_{t\in T(s,V_j^i(\alpha))} B_{Q-j+1}(t) \qquad (82)
\end{aligned}
$$

[00243]

[00244] The important factor is that the A_m and B_m needed can be calculated independently of i and α via U_Q and, respectively, U_{Q+1}. Above, $\tilde{A}_Q(W(s_0,V_{Q+1}))$ was formally determined via the auxiliary construct [~~which, however, is no longer needed in the resultant explicit representation.~~] $U_Q^i(\alpha)$ **which, however, is no longer needed in the resultant explicit representation.**

SUMMARY OF THE PROCEDURE:

- Define

$$
\begin{aligned}
V_j &:= V, && \text{for } j\in\{1,\ldots,a\}, \\
V_j &:= \{v_0\}, && \text{for } j\in\{a+1,\ldots,Q+1\}, \\
V_j^i(\alpha) &:= \{v\in V;\ v_{\hat{i}}=\alpha\}, && \text{for } i=(j-1)\cdot b+\hat{i}, \\
& && \hat{i}\in\{1,\ldots,b\}, \\
& && j\in\{1,\ldots,a\},\ \alpha\in\{\pm 1\}.
\end{aligned}
$$

[00245]

- For an arbitrary, but fixed choice of $y \in \mathbb{R}^N$, define for $q \in \{1,\ldots,Q\}$

$$
\begin{aligned}
&\mu_q : S \to \mathbb{R}, \\
&s \mapsto \exp\left(-\frac{1}{2\sigma^2}\sum_{j=1}^{n}\left(y_{n(q-1)+j} - C_j(s)\right)^2\right) = \exp\left(-\frac{1}{2\sigma^2}\Delta F_q(s)\right).
\end{aligned}
$$

[00246]

- Calculate

[00247]
$$A_m(s), \quad \text{for} \quad s \in S,\ m \in \{1, \ldots, a-1\},$$
$$B_m(s), \quad \text{for} \quad s \in S,\ m \in \{1, \ldots, Q\},$$

[00248] according to the recursion formulae (57) and (70) and starting values $A_0(s)$, $B_0(s)$, specified above, with (56) and (71).

- Calculate all A_α^i, $i \in \{1,\ldots,L\}$, $\alpha \in \{\pm 1\}$ over

[00249]
$$A_\alpha^i(y) = \sum_{s \in S} A_{j-1}(s) \sum_{t \in T(s, V_j^i(\alpha))} B_{Q-j+1}(t). \qquad (83)$$

[00250] and determine the soft outputs

[00251]
$$L(U_i|y) = \ln\left(\frac{A_{+1}^i(y)}{A_{-1}^i(y)}\right), \qquad i = 1, \ldots, K.$$

[00252] Together with the recursion formula from the preceding section, all $A_\alpha^i(y)$ can now be calculated jointly with $O(2^L \cdot Q)$ or, respectively, $O(K)$ operations instead of $O(K2^K)$ operations.

[00253] **Note that**

[00254] [~~Reminder:~~][~~$L = k$~~]~~Reminder:~~ $L = k \cdot$[~~$b, Q = a + k - 1, K = a$~~] $b, Q = a + k - 1, K = a \cdot$[~~$b,$~~] $b,$

[00255] where a is the number of information bits.

[00256] The numeric complexity for calculating the soft outputs has thus been reduced from an exponential order to a linear order where a, the number of information bits, is the decisive quantity.

SPECIAL CASE: BINARY STATE TRANSITION ($B = 1$)

[00257] In the important special case of $b = 1$, the set V of state transition symbols only consists of the two elements +1, -1. The GSM codes, for instance, belong to this widespread special case.

[00258] Since now $i = j$ and $V_j^i(\alpha) = \{\alpha\}$ in the above description, the procedure is simplified as follows:

- Define

$$V_j := \{\pm 1\}, \quad \text{for } j \in \{1, \dots, a\},$$
$$V_j := \{+1\}, \quad \text{for } j \in \{a+1, \dots, Q+1\}$$

[00259]

- For an arbitrary, but fixed choice of $y \in \mathrm{R}^N$ define for $q \in \{1,\dots,Q\}$

$$\mu_q : S \to \mathbb{R},$$
$$s \mapsto \exp\left(-\frac{1}{2\sigma^2}\sum_{j=1}^{n}\left(y_{n(q-1)+j} - C_j(s)\right)^2\right) = \exp\left(-\frac{1}{2\sigma^2}\Delta F_q(s)\right).$$

[00260]

- Calculate

$$A_m(s), \quad \text{for } s \in S, \ m \in \{1, \dots, a-1\},$$
$$B_m(s), \quad \text{for } s \in S, \ m \in \{1, \dots, Q\},$$

[00261]

[00262] according to the recursion formulae (57) and (70) and starting values $A_0(s)$, $B_0(s)$ with (56) and (71).

- Calculate all A_α^i, $i \in \{1,\dots,K\}$, $\alpha \in \{\pm 1\}$ over

$$A_\alpha^i(y) = \sum_{s \in S} A_{i-1}(s) B_{Q-i+1}(T(s, \alpha)). \qquad (84)$$

[00263]

[00264] and determine the soft outputs

$$L(U_i|y) = \ln\left(\frac{A_{+1}^i(y)}{A_{-1}^i(y)}\right), \quad i = 1, \dots, K.$$

[00265]

ALGORITHMIC CONVERSION

[00266] For the algorithmic conversion, consider the trellis diagram

$$\mathcal{T} = \{(s,q);\ s \in S,\ q = 0,\ldots,Q+1\}$$

[00267]

[00268] and the mappings

- node weights in state *s* of trellis segment *q*

$$\mu :\ \mathcal{T} \to \mathbb{R},$$
$$(s,q) \mapsto \exp\left(-\frac{1}{2\sigma^2}\Delta F_q(s)\right)$$

[00269]

- Subtotals 'A' in state *s* of trellis segment *q*

$$A :\ \mathcal{T} \to \mathbb{R},$$
$$(s,q) \mapsto A(s,q)$$

[00270]

- Subtotals 'B' in state *s* of trellis segment *Q*-*q*+1

$$B :\ \mathcal{T} \to \mathbb{R},$$
$$(s,q) \mapsto B(s,q)$$

[00271]

[00272] The mappings are only evaluated in the meaningful subsets of the definition domain.

[00273] Figure 2 shows an algorithm in pseudocode notation which represents a progression in the trellis diagram, considering all states for the calculation of the node weights. The algorithm illustrates the above statements and is comprehensible out of itself. Since the value of $\Delta F_q(s)$ depends only indirectly on the state *s* and is formed directly with $C(s)$, the following holds true

$$|\{\Delta F_q(s);\ s \in S\}| \leq \min\{2^L, 2^n\},$$

[00274]

[00275] i.e., for $n < L$, many of the above $\mu(s,q)$ have the same value. Depending on the special code, $\mu(s,q)$ can thus be determined with far fewer operations in the implementation.

[00276] Figure 3 and Figure 4 each show an algorithm in pseudocode notation for determining soft outputs. Figure 3 relates to the general case and Figure 4 relates to the special case for the binary state transition ($b = 1$). Both algorithms illustrate the above statements and are comprehensible [~~out~~]**<u>in and</u>** of themselves.

[00277] With a suitable implementation representation of V and, respectively, $V_j^i(\alpha)$, for instance as subsets of N, the above iterations $v \in V$ and $s \in S$ can be implemented as normal program loops. Naturally, indices which may occur such as, for example, $k - 1 + q$, are calculated only once in the implementation and not with every occurrence as is written down here for better clarity.

[00278] Figure 5 shows a processor unit PRZE. The processor unit PRZE comprises a processor CPU, a memory SPE and an input/output interface IOS which is used in various ways via an interface IFC: an output can be displayed on a monitor MON and/or output on a printer PRT via a graphics interface. An input is made via a mouse MAS or a keyboard TAST. The processor unit PRZE also has a data bus BUS which ensures the connection of a memory MEM, the processor CPU and the input/output interface IOS. Furthermore, additional components, for example an additional memory, data store (hard disk) or scanner, can be connected to the data bus BUS.

[00279] **<u>The above-described method and apparatus are illustrative of the principles of the present invention. Numerous modifications and adaptations will be readily apparent to those skilled in this art without departing from the spirit and scope of the present invention.</u>**

ABSTRACT

METHOD AND ARRANGEMENT FOR DECODING A PREDETERMINED CODE WORD

[00280] A method for decoding a predetermined code word is specified in which the code word comprises a number of positions having different values. In this method, encoding is performed, in particular, by [~~means~~]**way** of a terminated convolutional code. Each position of the code word is correlated with a safety measure (soft output) for a most probable Boolean value by performing the correlation on the basis of a trellis representation. The decoding of the code word is determined by the correlation of the individual positions of the code word.

4/PRTS

09/856936

GR 98 P5854

531 Rec'd PCT/ 30 MAY 2001

Description

Method and arrangement for decoding a predetermined code word

The invention relates to a method and an arrangement for decoding a predetermined code word.

In the decoding of a code word which has a predetermined number of positions, the information-carrying positions are to be restored as completely as possible.

The decoding takes place at the end of the receiver which has received the code word via a disturbed channel. Signals are transmitted, in particular as Boolean values, preferably subdivided into +1 and -1, via the channel where they are subject to a disturbance, and are converted into analog values which can deviate to a greater or lesser extent from the predetermined Boolean values (±1) by a demodulator.

The general assumption is K positions of binary information ("information bits") without redundancy $u \in \{\pm 1\}^K$, which is mapped into a code word $c \in \{\pm 1\}^N$ by means of systematic block codes or unsystematic block codes by a channel coder. In this arrangement, the code word contains $N - K$ bits (also "check bits") which can be used as redundant information to the N information bits for restoring the information after transmission via the disturbed channel.

The systematic block code adds to the N information bits $N - K$ check bits which are calculated from the information bits, the information bits themselves remaining unchanged whereas, in the unsystematic block code, the information bits themselves are changed, for example the information is in an operation performed

from one to the next position. Here, too, check bits are provided for reconstructing the information hidden in the operations. In the text which follows, in particular, a technically significant variant of unsystematic block codes, the so-called terminated convolutional codes, is considered.

"Hard" decoding of a correlation of the received code word (with the positions occupied by analog values) i.e. correlating each position with the nearest Boolean value in each case, is a decisive disadvantage since valuable information is lost in this process.

It is the object of the invention to determine a decoding of a predetermined code word, with decoding supplying analog values

(so-called "soft outputs") which, in particular, can be taken into consideration in the subsequent decoding method and thus provide for high error correction in the transmission of code words via a disturbed channel.

This object is achieved in accordance with the features of the independent patent claims. Further developments of the invention are also obtained from the dependent claims.

To achieve the object, a method for decoding a predetermined code word is specified in which the code word comprises a number of positions having different values. In this arrangement, encoding has taken place, in particular, by means of a terminated convolutional code. Each position of the code word is correlated with a safety measure (soft output) for a most probable Boolean value by performing the correlation on the basis of a trellis representation. The decoding of the code word is determined by the correlation of the individual positions of the code word.

A decisive advantage is here that due to the correlation based on the trellis representation, a distinct reduction in complexity compared with a general representation takes place with the result that decoding of the code word (generation of the soft outputs at the positions of the code word) also becomes possible in real time.

A further development consists in that the decoding rule for each position of the code word is determined by

$$L(U_i|y) = \ln\left(\frac{\sum\limits_{c\in\Gamma^i(+1)} \exp\left(-\frac{(y-c)^T(y-c)}{2\sigma^2}\right)}{\sum\limits_{c\in\Gamma^i(-1)} \exp\left(-\frac{(y-c)^T(y-c)}{2\sigma^2}\right)}\right), \quad \text{for } i = 1,\ldots,K, \qquad (1)$$

where

$L(U_i|y)$ is a safety measure (soft output) for the i-th position of the code word to be determined;

y is a demodulation result to be decoded;

c is a code word;

$\Gamma^i(\pm 1)$ are all code words for $u_i=\pm 1$;

σ^2 is a variance (channel disturbance).

Another further development consists in that the equation (1) is solved by utilizing a characteristic of a convolutional code used in the coding (and correspondingly in the decoding) which determines states in accordance with

a shift register operation used during the convolution, from which states, in turn, the trellis representation is obtained.

In an additional further development, the trellis representation is run through in a predetermined direction in order to recursively calculate terms A_m and A_m respectively. Into this calculation rule, node weights $\mu_m(s)$ which are determined by the demodulation result y enter at the nodes (s, m) of the trellis representation. The terms A_m and A_m are described by

$$\tilde{A}_m(E) = \sum_{s \in E} A_m(s), \quad \text{for } m \in \mathbb{N} \tag{2}$$

with

$$A_m(s) = \mu_m(s) \sum_{t \in W(s, V_m)} A_{m-1}(t), \quad \text{for } m \in \mathbb{N} \tag{3}$$

and a starting value

$$A_0(s) = \begin{cases} 1 & : \text{ for } s = s_0, \\ 0 & : \text{ else} \end{cases} \tag{4}$$

A more detailed discussion of the forms of description listed here can also be found in the description of the exemplary embodiment.

One embodiment consists in that mappings B_m are determined by means of the trellis representation, the trellis representation being processed in opposition to the predetermined direction. The term B_m is determined by

$$B_m(s) = \mu_{Q-m+1}(s) \sum_{t \in T(s, V_{Q-m+2})} B_{m-1}(t), \qquad \text{for } 1 \leq m \leq Q, \tag{5}$$

where

$$B_0(s) = \begin{cases} 1 & : \text{ for } s = s_0, \\ 0 & : \text{ else} \end{cases} \tag{6}$$

is determined for terminating the recursion.

Furthermore, terms A_α^i can be determined by again running through the trellis representation taking into consideration the terms A_m and B_m already determined. In particular, the terms A_α^i are determined in accordance with

$$A_\alpha^i(y) = \sum_{s \in S} A_{j-1}(s) \sum_{t \in T(s, V_j^i(\alpha))} B_{Q-j+1}(t). \tag{7}$$

In a further embodiment, the K positions of the decoded code word are determined in accordance with

$$L(U_i|y) = \ln\left(\frac{A^i_{+1}(y)}{A^i_{-1}(y)}\right), \quad i = 1, \ldots, K. \tag{8}$$

In particular, an AWGN (Additive Gaussian White Noise) channel model is used for the derivation. The method presented can also be used for other channel models, especially for channel models used in mobile radio.

Another embodiment relates to the use of the method in a mobile radio network, especially the GSM network.

It is also a further development that, after the soft outputs have been determined, there is a "hard" correlation of the analogue values with the Boolean values ±1. In this arrangement, the nearest Boolean value is in each case determined for correlating the analogue value.

The soft output values determined can be used as input values for further decoding when concatenated codes are used.

To achieve the object, an arrangement for decoding a predetermined code word is also specified in which a processor unit is provided which is set up in such a manner that

1. the code word comprises a number of positions having different values;

2. each position of the code word can be correlated with a soft output value by performing the correlation on the basis of a trellis representation;

3. the decoding of the code word can be determined by the correlation of the individual positions of the code word.

This arrangement is particularly suitable for performing the method according to the invention or one of its further developments explained above.

In the text which follows, exemplary embodiments of the invention will be shown and explained with reference to the drawing, in which:

Figure 1 shows a representation of digital information transmission;

Figure 2 shows an algorithm in pseudocode notation for progressing in the trellis diagram observing all states for the calculation of node weights;

Figure 3 shows an algorithm in pseudocode notation for determining soft outputs (general case);

Figure 4 shows an algorithm in pseudocode notation for determining soft outputs (special case: binary state transition);

Figure 5 shows a processor unit.

In the text which follows, first the convolutional code, then the reduction in complexity in the calculation of soft outputs and, finally, an algorithmic translation of the reduction in complexity are described in greater detail.

Terminated convolutional code

In communication technology, terminated convolutional codes are mostly used in concatenation with other systematic or unsystematic block codes. In particular, the decoding result of a convolutional decoder is used as the input for another decoder.

To ensure the lowest possible error rate, it is necessary to supply "soft" decoding decisions instead of "hard" ones in the convolutional decoding for the further decoder, i.e. to generate a tuple of "soft" values (soft outputs) from R instead of a tuple of "hard" Boolean (±1) values. The absolute value of the respective "soft" decision then provides a safety

measure for the correctness of the decision.

In principle, these soft outputs can be calculated in accordance with equation (1), depending on the channel model. However, the numeric complexity for calculating a soft output is $O(2^K)$, where K specifies the number of information bits. If K is realistically large, these formulae can thus not be evaluated, in particular, since such a code word must be calculated again every few milliseconds (real-time requirement).

One consequence was that soft outputs were dispensed with (with all consequences for the word and bit error rates) or, respectively, fewer elaborate approximations were performed for determining the soft outputs.

In the text which follows, a possibility for terminated convolutional codes is specified with the aid of which this complexity can be reduced to $O(K)$ in a trellis representation for calculating all soft outputs, i.e. it provides the possibility for a precise evaluation of equation (1).

In the text which follows, the bits of the code are represented in $\{\pm 1\}$ representation. In comparison with a $\{0, 1\}$ representation, which is often used in information technology, -1 corresponds to 1 and 1 corresponds to 0.

On a body $\{\pm 1\}$, addition $\oplus$ and multiplication $\odot$ are defined as follows:

$$\begin{array}{rcrcr} -1 & \oplus & -1 & = & 1 \\ -1 & \oplus & 1 & = & -1 \\ 1 & \oplus & -1 & = & -1 \\ 1 & \oplus & 1 & = & 1 \end{array} \qquad \begin{array}{rcrcr} -1 & \odot & -1 & = & -1 \\ -1 & \odot & 1 & = & 1 \\ 1 & \odot & -1 & = & 1 \\ 1 & \odot & 1 & = & 1 \end{array}$$

The coding is done with the aid of a "shift register" into which bit blocks (input blocks) of the information bits are written with each clock pulse. The combination of the bits of the shift register then generates one bit block of the code word. The shift register is pre-assigned +1 bits in each case. To terminate the coding (termination) blocks of tail zeros (+1) are shifted in afterwards. As has been mentioned initially, check bits by means of which bit errors can be corrected are correlated with the information bits by means of coding.

The following are defined for the further embodiments:

$b \in \mathrm{N}$	number of input bits per block
$V := \{\pm 1\}^b$	set of state transition signs
$a \in \mathrm{N}$	number of input blocks

$K := a \cdot b$	number of information bits without tail zeros
$k \in \mathrm{N},\ k \geq 2$	block length of the shift register, penetration depth
$L := k \cdot b$	bit length of the shift register
$S := \{\pm 1\}^L$	set of shift register signs
$n \in \mathrm{N}$	number of output bits per block
$Q := a + k - 1$	number of state transitions, input blocks + zeros
$N := n \cdot Q$	number of code bits
$R := \frac{b}{n}$	code rate

It should be noted here that the code rate is not K/N since the information bits have been counted without the zeros (+1) of the convolutional termination.

Furthermore, $s_0 \in S$ and $v_0 \in V$ are assumed to be the respective zero elements, i.e.

$$s_0 = (+1, \ldots, +1)^\mathsf{T}, \quad v_0 = (+1, \ldots, +1)^\mathsf{T}. \tag{9}$$

The state transition function of the shift register is assumed to be

$$T: \; S \times V \to S, \tag{10}$$
$$(s, v) \mapsto (s^{b+1}, \ldots, s^L, v^1, \ldots, v^b)^\mathsf{T}. \tag{11}$$

The terminated convolutional code is defined by the characterizing subsets

$$M_1, \ldots, M_n \subseteq \{1, \ldots, L\}, \tag{12}$$

(combination of register bits, alternatively in polynomial representation).

The current register content is coded via

$$C: \; S \to \{\pm 1\}^n, \tag{13}$$
$$s \mapsto C(s) \text{ where } C_j(s) := \bigoplus_{i \in M_j} s^i, \quad \text{for} \quad 1 \le j \le n. \tag{14}$$

where s^i is the i-th component of s.

Finally, the coding of an information word is defined by means of

$$\varphi: \{\pm 1\}^K \to \{\pm 1\}^N, \tag{15}$$

$$u \mapsto \begin{pmatrix} C(s_1) \\ \vdots \\ (s_Q) \end{pmatrix}, \tag{16}$$

where $s_0 \in S$ is the zero state (zero element),

$$u = \begin{pmatrix} \nu_1 \\ \vdots \\ \nu_a \end{pmatrix}, \quad \nu_i \in V, \quad 1 \le i \le a, \tag{17}$$

$$\nu_i := v_0, \quad a+1 \le i \le Q, \tag{18}$$

and furthermore

$$s_i := T(s_{i-1}, \nu_i), \quad 1 \le i \le Q. \tag{19}$$

According to the definition of T, the following is obtained

$$s_{Q+1} := T(s_Q, v_0) = s_0. \qquad (20)$$

Accordingly, the set of all code words is

$$\varphi\left(\{\pm 1\}^K\right) := \left\{\varphi(u) \in \{\pm 1\}^N;\ u \in \{\pm 1\}^K\right\}. \qquad (21)$$

Often, polynomials

$$p_j \in \{0,1\}[D] \quad \text{where} \quad \deg(p_j) \leq L-1$$

are used instead of the sets M_j for code definition, i.e.

$$p_j(D) = \sum_{i=0}^{L-1} \gamma_{i,j} D^i, \qquad (22)$$

with

$$\gamma_{i,j} \in \{0,1\} \qquad i = 0, \ldots, L-1, \quad j = 1, \ldots, n.$$

The following transformations then apply for $j = 1, \ldots, n$:

$$M_j = \{i \in \{1, \ldots, L\};\ \gamma_{L-i,j} = 1\} \qquad (23)$$

$$p_j(D) = \sum_{i \in M_j} D^{L-i}. \qquad (24)$$

Block code representation

Since a terminated convolutional code is a block code, the code bits c_j, $1 \leq j \leq N$ can also be represented from the information bits u_i, $1 \leq i \leq K$, with index sets J_j, as follows:

$$c_j := \bigoplus_{i \in J_j} u_i, \quad \text{for } 1 \leq j \leq N, \tag{25}$$

where

$$J_1, \ldots, J_N \subseteq \{1, \ldots, K\}. \tag{26}$$

The index sets J_j can be calculated directly from the above index sets M_m of the code definition.

Consider

$$j = n(q-1) + m, \quad q = 1, \ldots, Q, \; m = 1, \ldots, n. \tag{27}$$

$$c_j = C_m(s_q) = \bigoplus_{i \in M_m} (s_q)^i = \bigoplus_{i \in M_m} u_{i+b(q-k)}, \tag{28}$$

where $u_i := +1$ for $i \notin \{1, \ldots, K\}$.

Furthermore,

$$c_j \overset{\vee}{=} \bigoplus_{i-b(q-k) \in M_m} u_i = \bigoplus_{i \in M_m + b(q-k)} u_i, \tag{29}$$

and it thus follows for $j = 1, \ldots, N$ that

$$\begin{aligned} J_j &= \{1, \ldots, K\} \cap (M_m + b(q-k)) \\ &= \{i \in \{1, \ldots, K\};\; i - b(q-k) \in M_m\}. \end{aligned} \tag{30}$$

Example: SACCH convolutional code

In the above terminology, the convolutional code described in section 4.1.3 of the GSM Technical Specification GSM 05.03, Version 5.2.0 (channel coding) is:

$b = 1$	number of input bits per block
$V = \{\pm 1\}$	set of state transition signs
$a = 224$	number of input blocks
$K = 224$	number of information bits without tail zeros
$k = 5$	block length of the shift register, depth of penetration
$L = 5$	bit length of the shift register
$S = \{\pm 1\}^5$	set of shift register signs

$n = 2$	number of output bits per block
$Q = 228$	number of state transitions, input blocks + zeros
$N = 456$	number of code bits
$R = \frac{1}{2}$	code rate
$M_1 = \{1,2,5\}$	characterizing set; polynomial: $1+D^3+D^4$
$M_2 = \{1,2,4,5\}$	characterizing set; polynomial: $1+D+D^3+D^4$

Soft outputs in an AWGN channel model

In the text which follows, calculation rules for determining the soft outputs are derived, especially for the sake of clarity.

For this purpose, a probability space (Ω, S, P) and a K-dimensional random variable $U:\Omega\rightarrow\{\pm1\}^K$ are considered which have the properties

- The components $U_1, \ldots, U_K : \Omega\rightarrow\{\pm1\}$ are stochastically independent.
- The following holds for $i = 1, \ldots, K$

$$P(\{\omega \in \Omega;\ U_i(\omega) = -1\}) = P(\{\omega \in \Omega;\ U_i(\omega) = +1\}). \qquad (31)$$

Figure 1 shows a representation of digital telecommunication. A unit consisting of source 201, source encoder 202 and crypto-encoder 203 determines an information item $u \in \{\pm1\}^K$ which is used as input for one (or possibly more) channel encoder(s) 204. The channel encoder 204 generates a code word $c \in \{\pm1\}^N$ which is fed into a modulator 205 and is transmitted via a disturbed physical channel 206 to a receiver where it is determined to become a real-value code word $y \in R^N$ in a demodulator 207. This code word is converted into a real-value information item in a channel decoder 208. If necessary, a "hard" correlation with the Boolean values ±1 can also be made in a further decoder so that the received information is present in Boolean notation. The receiver is completed by a unit of crypto-decoder 209, source decoder 210 and sink 211. The two crypto-encoder 203 and crypto-decoder 209 units are optional in this arrangement.

The information to be reconstructed, $u \in \{\pm1\}^K$, of the crypto-encoder 203 is interpreted as implementation of

the random variables U since nothing is known about the choice of u in the receiver.

Thus, the output $c \in \{\pm 1\}^N$ of the channel encoder 204 is an implementation of the random variables $\varphi(U)$.

The output $y \in \mathbb{R}^N$ of the demodulator 207 is interpreted as implementation of the random variables

$$Y : \Omega \to \mathbb{R}^N, \tag{32}$$

$$\omega \mapsto \varphi(U(\omega)) + Z(\omega), \tag{33}$$

a random variable $Z : \Omega \to R^N$ representing the channel disturbances in the physical channel 206.

In the text which follows, an AWGN channel model is assumed, i.e. Z is a $N(0,\sigma^2 I_N)$ normally distributed random variable which is stochastically independent of U and, respectively, $\varphi(U)$. The variance σ^2 is calculated from the ratio between noise power density and mean energy in the channel 206 and is here assumed to be known.

The unknown output $u \in \{\pm 1\}^K$ of the crypto-encoder is to be reconstructed on the basis of an implementation y of Y. To estimate the unknown quantities $u_1, \ldots, u_K$, the distribution of the random variables U is investigated given the condition that y has been received.

The consequence of the fact that the random variable Y is a steady random variable is that the consideration of U under the condition that y has been received ($Y(\hat{\omega})=y$) is extremely complicated.

Firstly, the following is defined for $i \in \{1, \ldots, K\}$ and $\alpha \in \{\pm 1\}$

$$\Gamma^i(\alpha) := \left\{\varphi(u);\ u \in \{\pm 1\}^K;\ u_i = \alpha\right\}. \tag{34}$$

In a preparatory step, the following quantities are considered for $\epsilon > 0$, paying attention to the injectivity of the coding map φ:

$$\begin{aligned} L_\epsilon(U_i|y) &:= \ln\left(\frac{P(\{\omega \in \Omega;\ U_i(\omega) = +1\} \mid \{\omega \in \Omega;\ Y(\omega) \in M_{y,\epsilon}\})}{P(\{\omega \in \Omega;\ U_i(\omega) = -1\} \mid \{\omega \in \Omega;\ Y(\omega) \in M_{y,\epsilon}\})}\right) \\ &= \ln\left(\frac{\sum_{c\in\Gamma^i(+1)} P(\{\omega \in \Omega;\ \varphi(U(\omega)) = c\} \mid \{\omega \in \Omega;\ Y(\omega) \in M_{y,\epsilon}\})}{\sum_{c\in\Gamma^i(-1)} P(\{\omega \in \Omega;\ \varphi(U(\omega)) = c\} \mid \{\omega \in \Omega;\ Y(\omega) \in M_{y,\epsilon}\})}\right), \end{aligned} \tag{35}$$

for $i=1,\ldots,K$, where $M_{y,\varepsilon} := [y_1, y_1+\varepsilon] \times \ldots \times [y_N, y_N+\varepsilon]$.

Using the theorem by Bayes, the following is obtained:

$$
\begin{aligned}
L_\epsilon(U_i|y) &= \ln\left(\frac{\sum\limits_{c\in\Gamma^\prime(+1)} P(\{\omega\in\Omega;\ Y(\omega)\in M_{y,\epsilon}\} \mid \{\omega\in\Omega;\ \varphi(U(\omega))=c\})}{\sum\limits_{c\in\Gamma^\prime(-1)} P(\{\omega\in\Omega;\ Y(\omega)\in M_{y,\epsilon}\} \mid \{\omega\in\Omega;\ \varphi(U(\omega))=c\})}\right) \\
&= \ln\left(\frac{\sum\limits_{c\in\Gamma^\prime(+1)} \int\limits_{M_{y,\epsilon}} \exp\left(-\frac{(x-c)^\top(x-c)}{2\sigma^2}\right)dx}{\sum\limits_{c\in\Gamma^\prime(-1)} \int\limits_{M_{y,\epsilon}} \exp\left(-\frac{(x-c)^\top(x-c)}{2\sigma^2}\right)dx}\right).
\end{aligned} \tag{36}
$$

Considering then the limiting process of $L_\varepsilon(U_i|y)$ for $\varepsilon \downarrow 0$ by using L'Hospital's rule several times, the soft output $L(U_i|y)$ is obtained for each symbol as in equation (1).

Since

$$\Gamma^i(+1) \cup \Gamma^i(-1) = \{\pm 1\}^K$$

holds true, a total of $O(2^K)$ numeric operations are necessary for evaluating equation (1).

The vector $L(U_\bullet|y) \in R^K$ is the result of decoder 208.

Reduction of complexity in the determination of the soft outputs

Soft-output determination for convolutional codes

Firstly, the special characteristics of the terminated convolutional coding are used for providing an organized representation of the soft-output formula (1).

For an arbitrary, but preselected output $y \in R^N$ of the demodulator 207, the following weighting function (a Viterbi metric) of code words is considered:

$$F : \{\pm 1\}^N \to \mathbb{R}_0^+, \tag{37}$$

$$c \mapsto \sum_{j=1}^{N} (y_j - c_j)^2. \tag{38}$$

For permissible code words $c \in \{\pm 1\}^N$, i.e. $c \in \varphi(\{\pm 1\}^K)$, $F(c)$ can be reduced as follows, using the shift register representation:

$$F(c) = \sum_{q=1}^{Q} \underbrace{\sum_{j=1}^{n} \left(y_{n(q-1)+j} - C_j(\tilde{s}_q^c)\right)^2}_{=:\Delta F_q(\tilde{s}_q^c)}, \qquad (39)$$

where $\tilde{s}_q^c$ stands for the q-th state of the shift register in the (unambiguous) generation of the word c.

Then the following is defined for $I=1,\ldots,K$ and $\alpha \in \{\pm 1\}$:

$$A_\alpha^i(y) := \sum_{c \in \Gamma^i(\alpha)} \exp\left(-\frac{(y-c)^T(y-c)}{2\sigma^2}\right) = \sum_{c \in \Gamma^i(\alpha)} \prod_{q=1}^{Q} \exp\left(-\frac{1}{2\sigma^2}\Delta F_q(\tilde{s}_q^c)\right). \tag{40}$$

Thus, the following holds true for the soft outputs

$$L(U_i|y) = \ln\left(\frac{A_{+1}^i(y)}{A_{-1}^i(y)}\right), \quad i = 1, \ldots, K. \tag{41}$$

In the text which follows, the values $A_\alpha^i(y)$ are determined with the aid of a trellis diagram representation (also: trellis diagram or trellis representation).

To reduce the complexity of calculation, the following procedure is adopted in the following sections:

- generalization of A_α^i by mappings $\tilde{A}_m$.
- Recursive representation of $\tilde{A}_m$ by mappings A_m, the values of which are calculated with a "from left to right" run through a trellis diagram.
- Reversal of the recursion by mappings B_m, the values of which are calculated with a "from right to left" run through a trellis diagram.
- joint calculation of all A_α^i by means of a further run through a trellis diagram by using A_m and B_m.

The trellis diagram is here a set

$$\mathcal{T} = \{(s, q);\ s \in \mathcal{S},\ q = 0, \ldots, Q+1\} \tag{42}$$

The elements *(s,q)* of this set are also called the nodes in the trellis diagram, *s* representing a state and *q* being considered as a dynamic value (especially time).

General recursive representation

Firstly, some definitions are needed for representing the A_α^i in a generalized form which allows later transformation. For this reason, the following is determined

$$s_1^u := T(s_0, u_1), \quad u \in V^m = V \times \ldots \times V,\ m \geq 1, \tag{43}$$
$$s_j^u := T(s_{j-1}^u, u_j) \quad u \in V^m,\ m \geq j > 2, \tag{44}$$

i.e., s_j^u represents the state of the shift register after j shifts of the register with the input symbols $u_1, \ldots, u_j$.

Furthermore, sets $V_j \subseteq V$, $j \in$ N, which contain the permissible state transition symbols in the j-th step, are considered. Furthermore, product sets are defined as

$$U_m := V_1 \times \ldots \times V_m \subseteq V^m, \quad m \in \mathbb{N}, \tag{45}$$

i.e. U_m contains the first m components of the permissible input words.

For $q \in$ N, mappings

$$\mu_q : S \to \mathbb{R} \tag{46}$$

are considered and for $m \in$ N and input word sets $U_m \subseteq V^m$, mappings are defined as follows

$$\tilde{A}_m : \wp(S) \to \mathbb{R}, \tag{47}$$
$$E \mapsto \sum_{\substack{(u \in U_m) \\ \wedge (s_m^u \in E)}} \prod_{j=1}^{m} \mu_j(s_j^u), \tag{48}$$

i.e. summing over all permissible input words, the shift register of which reaches a final state in E, is performed. If there are no such input words, the sum is determined as 0 over an empty index set.

In addition, a mapping is determined as

$$W : S \times \wp(V) \to \wp(S), \tag{49}$$

$$(t, \hat{V}) \mapsto \left\{ s \in S;\ \exists \hat{v} \in \hat{V} \ni T(s, \hat{v}) = t \right\}, \tag{50}$$

i.e., W maps $(t, \hat{V})$ into the sets of all states which can reach the state t with a transition symbol from $\hat{V}$.

The following holds true for $m \geq 2$, $E \subseteq S$

$$\begin{aligned}
\tilde{A}_m(E) &= \sum_{\substack{(u \in U_m) \\ \wedge (s_m^u \in E)}} \prod_{j=1}^{m} \mu_j(s_j^u) \\
&= \sum_{s \in E} \sum_{\substack{(u \in U_m) \\ \wedge (s_m^u = s)}} \prod_{j=1}^{m} \mu_j(s_j^u) \\
&= \sum_{s \in E} \mu_m(s) \sum_{\substack{(u \in U_m) \\ \wedge (s_m^u = s)}} \prod_{j=1}^{m-1} \mu_j(s_j^u) \\
&= \sum_{s \in E} \mu_m(s) \sum_{\substack{(u \in U_{m-1}) \\ \wedge (s_{m-1}^u \in W(s, V_m))}} \prod_{j=1}^{m-1} \mu_j(s_j^u) \\
&= \sum_{s \in E} \mu_m(s) \tilde{A}_{m-1}\left(W(s, V_m)\right). \qquad (51)
\end{aligned}$$

In the transformation in the last step but one, attention must be paid to the fact that there is **exactly one** transition symbol $v \in V_m$ with $T\left(s_{m-1}^u, v\right) = s$, if s_{m-1}^u is in $W(s, V_m)$, i.e. it is not necessary to take account of any multiplicities.

Consider, then, the following for $m \geq 2$ mappings

$$A_m : S \to \mathbb{R}, \qquad (52)$$

$$s \mapsto \mu_m(s) \tilde{A}_{m-1}\left(W(s, V_m)\right). \qquad (53)$$

Thus, a recursion formula can be derived for $m \geq 3$:

$$\begin{aligned}
A_m(s) &= \mu_m(s) \tilde{A}_{m-1}\left(W(s, V_m)\right) \\
&= \mu_m(s) \sum_{t \in W(s, V_m)} \mu_{m-1}(t) \tilde{A}_{m-2}\left(W(t, V_{m-1})\right) \\
&= \mu_m(s) \sum_{t \in W(s, V_m)} A_{m-1}(t). \qquad (54)
\end{aligned}$$

Furthermore:

$$
\begin{aligned}
A_2(s) &= \mu_2(s)\tilde{A}_1\left(W(s,V_2)\right) \\
&= \mu_2(s) \sum_{\substack{(u\in U_1) \\ \wedge(s_1^u\in W(s,V_2))}} \mu_1(s_1^u) \\
&= \mu_2(s) \sum_{t\in W(s,V_2)} \mu_1(t)\delta_{s_0\in W(t,V_1)} \\
&= \mu_2(s) \sum_{t\in W(s,V_2)} \underbrace{\mu_1(t) \sum_{\hat{t}\in W(t,V_1)} \underbrace{\delta_{\hat{t}=s_0}}_{=:A_0(\hat{t})}}_{=:A_1(t)}
\end{aligned}
\tag{55}
$$

In summary, the following thus holds true for $s \in S$, $E \subseteq S$:

$$
A_0(s) = \begin{cases} 1, & \text{for} \quad s = s_0, \\ 0, & \text{otherwise} \end{cases}, \tag{56}
$$

$$
A_m(s) = \mu_m(s) \sum_{t\in W(s,V_m)} A_{m-1}(t), \quad \text{for} \quad m \in \mathbb{N}, \tag{57}
$$

$$
\tilde{A}_m(E) = \sum_{s\in E} A_m(s), \quad \text{for} \quad m \in \mathbb{N}. \tag{58}
$$

The sets $W(s,V_m)$ can be represented constructively. For this purpose, two further mappings are considered. The following is defined

$$
\tau : S \to V, \tag{59}
$$

$$
s = (s^1,\ldots,s^L)^\top \mapsto (s^{L-b+1},\ldots,s^L)^\top, \tag{60}
$$

i.e. if the state s is the result of a state transition, then $\tau(s)$ was the associated state transition symbol.

Furthermore

$$\hat{T}: \; V \times S \to S, \tag{61}$$
$$(v, s) \mapsto (v^1, \ldots, v^b, s^1, \ldots, s^{L-b})^\mathsf{T}, \tag{62}$$

is defined, i.e. $\hat{T}$ reverses the direction of the shift register operation.

The following then holds

$$T\left(\hat{T}(v, s), \tau(s)\right) = s, \quad \textbf{for all} \quad s \in S, \; v \in V \tag{63}$$

and for all $t \in S$ and $\hat{V} \subseteq V$, it also holds true that

$$W(t,\hat{V}) = \left\{ s \in S;\ \exists \hat{v} \in \hat{V} \ni T(s,\hat{v}) = t \right\} = \begin{cases} \left\{ \hat{T}(v,t);\ v \in V \right\}, & \text{if } \tau(t) \in \hat{V}, \\ \emptyset, & \text{else.} \end{cases} \tag{64}$$

Thus, the recursion formula (57) for $A_m(s)$ can be written down constructively as follows:

$$A_m(s) = \mu_m(s) \sum_{t \in W(s,V_m)} A_{m-1}(t) = \begin{cases} \mu_m(s) \sum_{v \in V} A_{m-1}\left(\hat{T}(v,s)\right), & \text{if } \tau(s) \in V_m, \\ 0, & \text{else.} \end{cases} \tag{65}$$

It should be noted that in this section, no restrictions were set for the set V of the state transition symbols and for the sets $V_j \in \wp(V)$.

Reversal of recursion

In the text which follows, a recursion in the "reverse direction" compared with the above recursion is described. This new recursion is defined with the aid of the recursion formula (57) for $A_m(s)$.

The following is assumed for this purpose

$$T(t,\hat{V}) := \left\{ T(t,\hat{v});\ \hat{v} \in \hat{V} \right\}, \quad \text{for } t \in S,\ \hat{V} \subseteq V \tag{66}$$

and for $M \in N$, $0 \leq m \leq Q$, the following mappings are considered

$$B_m : S \rightarrow \mathbb{R}, \tag{67}$$

with the following recursive characteristic:

$$\begin{aligned}
&\sum_{s \in S} A_m(s) \sum_{t \in T(s,V_{m+1})} B_{Q-m}(t) = \\
&= \sum_{s \in S} \mu_m(s) \sum_{\hat{t} \in W(s,V_m)} A_{m-1}(\hat{t}) \sum_{t \in T(s,V_{m+1})} B_{Q-m}(t) \\
&= \sum_{\hat{t} \in S} \sum_{s \in T(\hat{t},V_m)} \mu_m(s) A_{m-1}(\hat{t}) \sum_{t \in T(s,V_{m+1})} B_{Q-m}(t) \\
&= \sum_{\hat{t} \in S} A_{m-1}(\hat{t}) \sum_{s \in T(\hat{t},V_m)} \underbrace{\mu_m(s) \sum_{t \in T(s,V_{m+1})} B_{Q-m}(t)}_{=: B_{Q-m+1}(s)},
\end{aligned}$$

i.e.

$$\sum_{s \in S} A_m(s) \sum_{t \in T(s,V_{m+1})} B_{Q-m}(t) = \sum_{s \in S} A_{m-1}(s) \sum_{t \in T(s,V_m)} B_{Q-m+1}(t). \qquad (68)$$

By applying equation (68) several times, the following is obtained for an arbitrary $j \in \{1,\ldots,m+1\}$

$$\sum_{s \in S} A_m(s) \sum_{t \in T(s,V_{m+1})} B_{Q-m}(t) = \sum_{s \in S} A_{j-1}(s) \sum_{t \in T(s,V_j)} B_{Q-j+1}(t). \qquad (69)$$

According to the above definition, the recursion formula is thus

$$B_m(s) = \mu_{Q-m+1}(s) \sum_{t \in T(s,V_{Q-m+2})} B_{m-1}(t), \qquad \text{for } 1 \le m \le Q. \qquad (70)$$

To terminate the recursion, the following are defined

$$B_0(s) = \begin{cases} 1, & \text{for } s = s_0, \\ 0, & \text{else} \end{cases} . \tag{71}$$

Given this termination and the equations (58) and (69),

$$\tilde{A}_Q(W(s_0, V_{Q+1}))$$

can be represented for $V_{Q+1} := \{v_0\}$ and with an arbitrary $j \in \{1,...,Q+1\}$ as follows:

$$
\begin{aligned}
\tilde{A}_Q(W(s_0, V_{Q+1})) &= \sum_{s \in W(s_0, V_{Q+1})} A_Q(s) \\
&= \sum_{s \in S} A_Q(s) \sum_{t \in T(s,\{v_0\})} B_0(t) \\
&= \sum_{s \in S} A_Q(s) \sum_{t \in T(s,V_{Q+1})} B_0(t) \\
&= \sum_{s \in S} A_{j-1}(s) \sum_{t \in T(s,V_j)} B_{Q-j+1}(t). \qquad (72)
\end{aligned}
$$

Note: in the evaluation of (72), V_j is not included in the calculation of the A_m and B_m needed.

Calculation of A_α^i

Using the preliminary work from the preceding sections, A_α^i can now be calculated in a simple manner.

For this purpose, the following are defined:

$$V_j := V, \quad \text{for} \quad j \in \{1, \ldots, a\}, \qquad (73)$$
$$V_j := \{v_0\}, \quad \text{for} \quad j \in \{a+1, \ldots, Q+1\}, \qquad (74)$$

i.e. all permissible code words are defined via the states s_j^u with

$$u \in U_Q = V_1 \times \ldots \times V_Q$$

The code words used in the calculation of the A_α^i are restricted by $u_i=\alpha$. For an arbitrary but fixed choice

of $i \in \{1,\ldots,K\}$, there is exactly one $j \in \{1,\ldots,a\}$ and exactly one $\hat{i} \in \{1,\ldots,b\}$ with

$$i = (j-1)\cdot b + \hat{i}. \qquad (75)$$

Furthermore, the following are defined for an arbitrary but fixed choice of $\alpha \in \{\pm 1\}$:

$$V_j^{i}(\alpha) := \{v \in V;\ v_{\hat{i}} = \alpha\} \qquad (76)$$

$$U_Q^{i}(\alpha) := V_1 \times \ldots \times V_{j-1} \times V_j^{i}(\alpha) \times V_{j+1} \times \ldots \times V_Q \subset U_Q, \qquad (77)$$

i.e. the code words from $\Gamma^i(\alpha)$ are determined via the states s_j^u with $u \in U_Q^i(\alpha)$.

For an arbitrary, but fixed choice of $y \in \mathrm{R}^N$, define for $q \in \{1,\ldots,Q\}$

$$\mu_q : S \to \mathbb{R}, \tag{78}$$

$$s \mapsto \exp\left(-\frac{1}{2\sigma^2}\sum_{j=1}^{n}\left(y_{n(q-1)+j} - C_j(s)\right)^2\right) = \exp\left(-\frac{1}{2\sigma^2}\Delta F_q(s)\right). \tag{79}$$

According to the definition of the convolutional code, the following holds true for all s_Q^u with $u \in U_Q$:

$$s_{Q+1}^u = T(s_Q^u, u_{Q+1}) = s_0, \quad u_{Q+1} \in V_{Q+1} = \{v_0\}, \tag{80}$$

i.e.

$$s_Q^u \in W(s_0, V_{Q+1}). \tag{81}$$

Taking account of equation (72), the following thus holds true:

$$\begin{aligned}
A_\alpha^i(y) &= \sum_{c\in\Gamma^i(\alpha)}\prod_{q=1}^{Q}\exp\left(-\frac{1}{2\sigma^2}\Delta F_q(\tilde{s}_q^c)\right)\\
&= \sum_{u\in U_Q^i(\alpha)}\prod_{q=1}^{Q}\mu_q(s_q^u)\\
&= \sum_{\substack{(u\in U_Q^i(\alpha))\\ \wedge(s_Q^u\in W(s_0,V_{Q+1}))}}\prod_{q=1}^{Q}\mu_q(s_q^u)\\
&= \tilde{A}_Q(W(s_0,V_{Q+1}))\\
&= \sum_{s\in S}A_{j-1}(s)\sum_{t\in T(s,V_j^i(\alpha))}B_{Q-j+1}(t)
\end{aligned} \tag{82}$$

The important factor is that the A_m and B_m needed can be calculated independently of i and α via U_Q and, respectively, U_{Q+1}. Above, $\widetilde{A}_Q(W(s_0, V_{Q+1}))$ was formally determined via the auxiliary construct $U_Q^i(\alpha)$ which, however, is no longer needed in the resultant explicit representation.

Summary of the procedure:

- Define

$$\begin{aligned} V_j &:= V, && \text{for} \quad j \in \{1, \ldots, a\}, \\ V_j &:= \{v_0\}, && \text{for} \quad j \in \{a+1, \ldots, Q+1\}, \\ V_j^i(\alpha) &:= \{v \in V;\ v_{\hat{i}} = \alpha\}, && \text{for} \quad i = (j-1) \cdot b + \hat{i}, \\ & && \quad\quad \hat{i} \in \{1, \ldots, b\}, \\ & && \quad\quad j \in \{1, \ldots, a\},\ \alpha \in \{\pm 1\}. \end{aligned}$$

- For an arbitrary, but fixed choice of $y \in R^N$, define for $q \in \{1, \ldots, Q\}$

$$\mu_q : S \to \mathbb{R},$$
$$s \mapsto \exp\left(-\frac{1}{2\sigma^2}\sum_{j=1}^{n}\left(y_{n(q-1)+j} - C_j(s)\right)^2\right) = \exp\left(-\frac{1}{2\sigma^2}\Delta F_q(s)\right).$$

- Calculate

$$\begin{aligned} &A_m(s), && \text{for} \quad s \in S,\ m \in \{1, \ldots, a-1\}, \\ &B_m(s), && \text{for} \quad s \in S,\ m \in \{1, \ldots, Q\}, \end{aligned}$$

according to the recursion formulae (57) and (70) and starting values $A_0(s)$, $B_0(s)$, specified above, with (56) and (71).

- Calculate all A_α^i, $i \in \{1, \ldots, L\}$, $\alpha \in \{\pm 1\}$ over

$$A_\alpha^i(y) = \sum_{s \in S} A_{j-1}(s) \sum_{t \in T(s, V_j^i(\alpha))} B_{Q-j+1}(t). \qquad (83)$$

and determine the soft outputs

$$L(U_i|y) = \ln\left(\frac{A^i_{+1}(y)}{A^i_{-1}(y)}\right), \quad i = 1, \ldots, K.$$

Together with the recursion formula from the preceding section, all $A^i_\alpha(y)$ can now be calculated jointly with $O(2^L \cdot Q)$ or, respectively, $O(K)$ operations instead of $O(K2^K)$ operations.

Reminder: $L = k \cdot b$, $Q = a + k - 1$, $K = a \cdot b$, where a is the number of information bits.

The numeric complexity for calculating the soft outputs has thus been reduced from an exponential order to a linear order where a, the number of information bits, is the decisive quantity.

Special case: Binary state transition (b = 1)

In the important special case of b = 1, the set V of state transition symbols only consists of the two elements +1, -1. The GSM codes, for instance, belong to this widespread special case.

Since now $i = j$ and $V_j^i(\alpha) = \{\alpha\}$ in the above description, the procedure is simplified as follows:

- Define

$$V_j := \{\pm 1\}, \quad \text{for } j \in \{1, \ldots, a\},$$
$$V_j := \{+1\}, \quad \text{for } j \in \{a+1, \ldots, Q+1\}$$

- For an arbitrary, but fixed choice of $y \in R^N$ define for $q \in \{1, \ldots, Q\}$

$$\mu_q : S \to \mathbb{R},$$
$$s \mapsto \exp\left(-\frac{1}{2\sigma^2}\sum_{j=1}^{n}\left(y_{n(q-1)+j} - C_j(s)\right)^2\right) = \exp\left(-\frac{1}{2\sigma^2}\Delta F_q(s)\right).$$

- Calculate

$$A_m(s), \quad \text{for } s \in S,\ m \in \{1, \ldots, a-1\},$$
$$B_m(s), \quad \text{for } s \in S,\ m \in \{1, \ldots, Q\},$$

according to the recursion formulae (57) and (70) and starting values $A_0(s)$, $B_0(s)$ with (56) and (71).

- Calculate all A_α^i, $i \in \{1, \ldots, K\}$, $\alpha \in \{\pm 1\}$ over

$$A_\alpha^i(y) = \sum_{s \in S} A_{i-1}(s) B_{Q-i+1}(T(s, \alpha)). \tag{84}$$

and determine the soft outputs

$$L(U_i | y) = \ln\left(\frac{A_{+1}^i(y)}{A_{-1}^i(y)}\right), \quad i = 1, \ldots, K.$$

Algorithmic conversion

For the algorithmic conversion, consider the trellis diagram

$$\mathcal{T} = \{(s,q);\ s \in S,\ q = 0,\dots,Q+1\}$$

and the mappings

- node weights in state s of trellis segment q

$$\begin{aligned}\mu:\ \mathcal{T} &\rightarrow \mathbb{R},\\ (s,q) &\mapsto \exp\left(-\frac{1}{2\sigma^2}\Delta F_q(s)\right)\end{aligned}$$

- Subtotals 'A' in state s of trellis segment q

$$\begin{aligned}A:\ \mathcal{T} &\rightarrow \mathbb{R},\\ (s,q) &\mapsto A(s,q)\end{aligned}$$

- Subtotals 'B' in state s of trellis segment $Q-q+1$

$$\begin{aligned}B:\ \mathcal{T} &\rightarrow \mathbb{R},\\ (s,q) &\mapsto B(s,q)\end{aligned}$$

The mappings are only evaluated in the meaningful subsets of the definition domain.

Figure 2 shows an algorithm in pseudocode notation which represents a progression in the trellis diagram, considering all states for the calculation of the node weights. The algorithm illustrates the above statements and is comprehensible out of itself. Since the value of $\Delta F_q(s)$ depends only indirectly on the state s and is formed directly with $C(s)$, the following holds true

$$\left|\{\Delta F_q(s);\ s \in S\}\right| \leq \min\left\{2^L, 2^n\right\},$$

i.e., for $n < L$, many of the above $\mu(s,q)$ have the same value. Depending on the special code, $\mu(s,q)$ can thus be determined with far fewer operations in the implementation.

Figure 3 and Figure 4 each show an algorithm in pseudocode notation for determining soft outputs. Figure 3 relates to the general case and Figure 4 relates to the special case for the binary state transition ($b = 1$). Both

algorithms illustrate the above statements and are comprehensible out of themselves.

With a suitable implementation representation of V and, respectively, $V_j^i(\alpha)$, for instance as subsets of N, the above iterations $v \in V$ and $s \in S$ can be implemented as normal program loops. Naturally, indices which may occur such as, for example, $k - 1 + q$, are calculated only once in the implementation and not with every occurrence as is written down here for better clarity.

Figure 5 shows a processor unit PRZE. The processor unit PRZE comprises a processor CPU, a memory SPE and an input/output interface IOS which is used in various ways via an interface IFC: an output can be displayed on a monitor MON and/or output on a printer PRT via a graphics interface. An input is made via a mouse MAS or a keyboard TAST. The processor unit PRZE also has a data bus BUS which ensures the connection of a memory MEM, the processor CPU and the input/output interface IOS. Furthermore, additional components, for example an additional memory, data store (hard disk) or scanner, can be connected to the data bus BUS.

Patent Claims

1. A method for decoding a predetermined code word,
   (a) in which the code word comprises a number of positions having different values;
   (b) in which each position of the code word is correlated with a soft-output value, in which the calculation rule for the soft-output value for each position of the code word is determined by

$$L(U_i|y) = \ln\left(\frac{\sum\limits_{c\in\Gamma^i(+1)} \exp\left(-\frac{(y-c)^\top(y-c)}{2\sigma^2}\right)}{\sum\limits_{c\in\Gamma^i(-1)} \exp\left(-\frac{(y-c)^\top(y-c)}{2\sigma^2}\right)}\right), \quad \text{for} \quad i = 1,\ldots,K\ ,$$

   where

$L(U_i\|y)$	is a safety measure (soft output) for the i-th position of the code word to be determined;
y	is a demodulation result to be decoded;
c	is a code word;
$\Gamma^i(\pm 1)$	are all code words for $u_i = \pm 1$;
σ^2	is a variance (channel disturbance);

   (c) in which the decoding of the code word is determined by the correlation of the individual positions of the code word by utilizing a characteristic of a convolutional code from which it follows that states are determined in accordance with a shift register operation, from which states a trellis representation is obtained;
   (d) in which, for an arbitrary choice of $y \in R^N$, weights $\mu_q(s)$ are calculated for the nodes $(s,\ q)$ of the trellis representation by evaluating

$$\mu_q : S \to \mathbb{R},$$
$$s \mapsto \exp\left(-\frac{1}{2\sigma^2}\sum_{j=1}^{n}\left(y_{n(q-1)+j} - C_j(s)\right)^2\right)$$

for $q \in \{1, \ldots, Q\}$.

(e) in which mappings A_m are determined by means of the trellis representation, running through the trellis representation in the natural direction, the term A_m being determined by

$$A_m(s) = \mu_m(s) \sum_{t \in W(s,V_m)} A_{m-1}(t), \quad \text{for } m \in \mathbb{N}$$

and a starting value

$$A_0(s) = \begin{cases} 1 & : \text{ for } s = s_0, \\ 0 & : \text{ else} \end{cases}$$

(f) in which mappings B_m are determined by means of the trellis representation, the trellis representation being run through in opposition to the predetermined direction, the term B_m being determined by

$$B_m(s) = \mu_{Q-m+1}(s) \sum_{t \in T(s, V_{Q-m+2})} B_{m-1}(t), \quad \text{for} \quad 1 \leq m \leq Q,$$

where

$$B_0(s) = \begin{cases} 1 & : \text{ for } s = s_0, \\ 0 & : \text{ else} \end{cases}$$

is determined for terminating the recursion;

(g) in which terms A_α^i are determined by again running through the trellis representation taking into consideration the terms A_m and B_m already determined, in accordance with the relation

$$A_\alpha^i(y) = \sum_{s \in S} A_{j-1}(s) \sum_{t \in T(s, V_j^i(\alpha))} B_{Q-j+1}(t),$$

where $j = \left\lfloor \frac{i-1}{b} \right\rfloor + 1$;

(h) in which the K positions of the code word are determined in accordance with

$$L(U_i|y) = \ln\left(\frac{A^i_{+1}(y)}{A^i_{-1}(y)}\right), \quad i = 1, \ldots, K.$$

2. The method as claimed in one of the preceding claims,

(a) in which the convolutional code has binary state transitions;

(b) in which the mappings A_m are determined recursively by

$$A_m(s) = \mu_m(s)\left(A_{m-1}\left(\hat{T}(+1,s)\right) + A_{m-1}\left(\hat{T}(-1,s)\right)\right), \quad \text{for} \quad m \in \mathbb{N};$$

(c) in which the mappings B_m are determined recursively by

$$B_m(s) = \mu_{Q-m+1}(s)\left(B_{m-1}\left(T(s,+1)\right) + B_{m-1}\left(T(s,-1)\right)\right),$$

for $1 \leq m \leq Q$;

(d) in which the terms A_α^i, $i \in \{1,\ldots,K\}$, $\alpha \in \{\pm 1\}$ are determined in accordance with

$$A_\alpha^i(y) = \sum_{s \in S} A_{i-1}(s) B_{Q-i+1}(T(s,\alpha)).$$

3. The method as claimed in one of the preceding claims, for use in a mobile radio network.

4. The method as claimed in claim 3, in which the mobile radio network is a GSM network.

5. The method as claimed in one of the preceding claims, in which the method is a part of the decoding of a concatenated code in which the calculated soft-output values are used as input data of another decoder.

6. An arrangement for decoding a predetermined code word, in which a processor unit is provided which is set up in such a manner that a method according to one of the preceding claims can be carried out by this unit.

GR 98 P5854

Abstract

Method and arrangement for decoding a predetermined code word

A method for decoding a predetermined code word is specified in which the code word comprises a number of positions having different values. In this method, encoding is performed, in particular, by means of a terminated convolutional code. Each position of the code word is correlated with a safety measure (soft output) for a most probable Boolean value by performing the correlation on the basis of a trellis representation. The decoding of the code word is determined by the correlation of the individual positions of the code word.

FIG 1

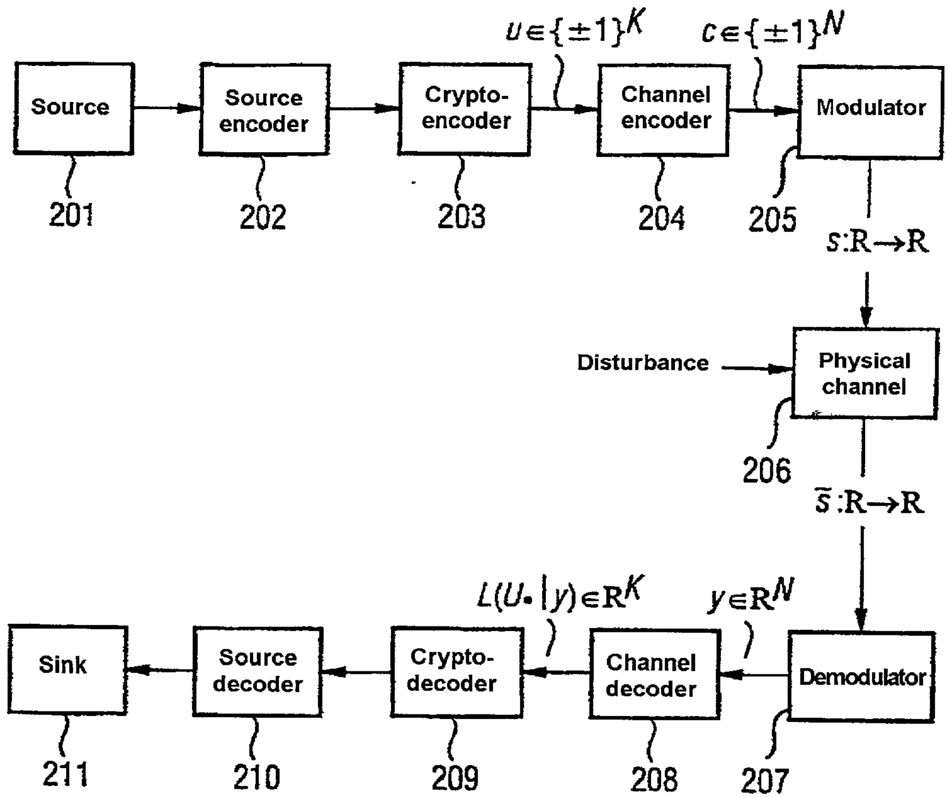


FIG 2

for $q = 1, \ldots, Q$:
  for $s \in S$:
    $\mu(s,q) := \exp\left(\frac{-1}{2\sigma^2} \Delta F_q(s)\right);$

FIG 3

for $s \in S$: — Pre-allocation

  $A(s,0 := 0; B(s,0) := 0;$

$A(s,0 := 1; B(s_0,0) := 1;$ — Starting state

Termination

for $q = 1,\ldots,k$: — Consideration of all states

  für $s \in S$: — Follow-up state

    $s^+ := T(s, v0);$ — Calculation of B

    $B(s,q) := \mu(s, Q-(q-1)) B(s^+, q-1);$

for $q = 1,\ldots,a-1$: — Progression in the trellis diagram

  für $s \in S$: — Consideration of all states

    $A(s,q) := A(\hat{T}(v_0,s),q-1);$ — Pre-allocation of A

    $B(s,k+q) := B(T(s,v_0),k-1+q);$ — Pre-allocation of B

    für $v \in V \setminus \{v_0\}$ — Consideration of all transitions

      $A(s,q) := A(s,q) + A(\hat{T}(v,s),q-1);$ — Calculation of A

      $B(s,k+q) := B(s,k+q) + B(T(s,v_0),k-1+q);$ — Calculation of B

    $A(s,q) := \mu(s,q) \cdot A(s,q);$ — Calculation of A

    $B(s,k+q) := \mu(s,a-q) \cdot B(s,k+q);$ — Calculation of B

for $i = 1,\ldots,K$: — Progression in the trellis diagram

  $A^i_{+1} := 0; A^i_{-1} := 0;$ — Pre-allocation

  $j = 1 + [(i-1)/b];$

  for $s \in S$: — Consideration of all states

    for $v \in V^i_j(+1)$: — Transitions

      $A^i_{+1} := A^i_{+1} + A(s,j-1) \cdot B(T(s,v),Q-j+1);$ — Update of $A^i + 1$

    for $v \in V^i_j(-1)$: — Transitions

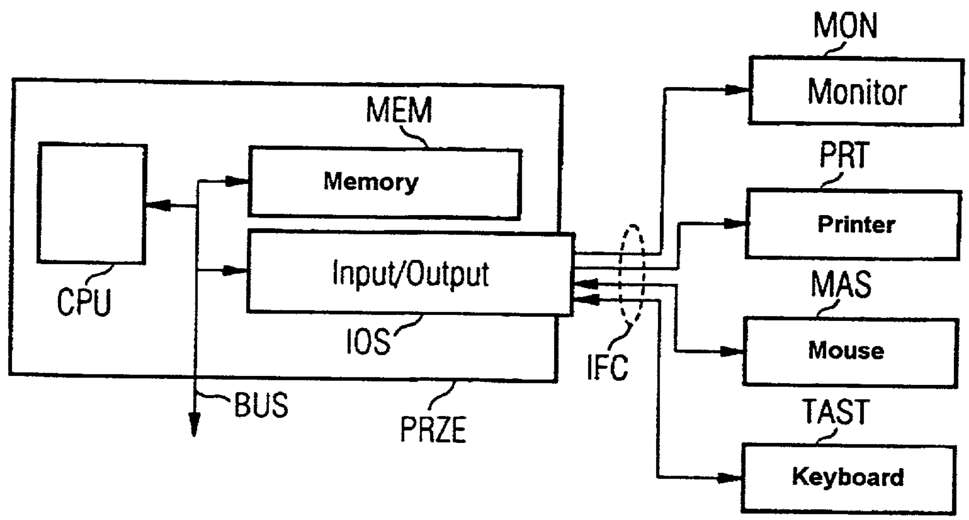
      $A^i_{-1} := A^i_{-1} + A(s,j-1) \cdot B(T(s,v),Q-j\mp 1);$ — Update of $A^i + 1$

  $L(U_i | y) := \ln(A^i_{+1}/A^i_{-1});$ — i-th soft output

FIG 4

for $s \in S$:	Pre-allocation
$\quad A(s,0 := 0; B(s,0) := 0;$	
$A(s_0,0 := 1; B(s_0,0) := 1;$	Starting state
for $q = 1,\ldots,k$:	Termination
$\quad$ for $s \in S$:	Consideration of all states
$\quad\quad s^+ := T(s,+1);$	Follow-up state
$\quad\quad B(s,q) := \mu(s,Q-(q-1))B(s^+,q-1);$	Calculation of B
for $q = 1,\ldots,a-1$:	Progression in the trellis diagram
$\quad$ for $s \in S$:	Consideration of all states
$\quad\quad t^+ := \hat{T}(+1,s); t^- := \hat{T}(-1,s);$	Predecessor states
$\quad\quad s^+ := T(s,+1); s^- := T(s,-1);$	Follow-up states
$\quad\quad A(s,q) := \mu(s,q)\cdot(A(t^+,q-1)+A(t^-,q-1));$	Calculation of A
$\quad\quad B(s,k+q) := \mu(s,a-q)\cdot(B(s^+,k-1+q)+B(s^-,k-1+q));$	Calculation of B
for $i = 1,\ldots,a$:	Progression in the trellis diagram
$\quad A^i_{+1} := 0; A^i_{-1} := 0;$	Pre-allocation
$\quad$ for $s \in S$:	Consideration of all states
$\quad\quad s^+ := T(s,+1); s^- := T(s,-1);$	Follow-up states
$\quad\quad A^i_{+1} := A^i_{+1} + A(s,i-1)\cdot B(s^+, Q-i+1);$	Update of A^i + 1
$\quad\quad A^i_{-1} := A^i_{-1} + A(s,i-1)\cdot B(s^-, Q-i+1);$	Update of A^i + 1
$\quad L(U_i \mid y) := \ln(A^i_{+1}/A^i_{-1});$	i-th soft output

FIG 5


MON
Monitor
MEM
Memory
PRT
Printer
CPU
Input/Output
MAS
Mouse
IOS
IFC
BUS
PRZE
TAST
Keyboard

Declaration and Power of Attorney For Patent Application
Erklärung Für Patentanmeldungen Mit Vollmacht
German Language Declaration

Als nachstehend benannter Erfinder erkläre ich hiermit an Eides Statt:

As a below named inventor, I hereby declare that:

dass mein Wohnsitz, meine Postanschrift, und meine Staatsangehörigkeit den im Nachstehenden nach meinem Namen aufgeführten Angaben entsprechen,

My residence, post office address and citizenship are as stated below next to my name,

dass ich, nach bestem Wissen der ursprüngliche, erste und alleinige Erfinder (falls nachstehend nur ein Name angegeben ist) oder ein ursprünglicher, erster und Miterfinder (falls nachstehend mehrere Namen aufgeführt sind) des Gegenstandes bin, für den dieser Antrag gestellt wird und für den ein Patent beantragt wird für die Erfindung mit dem Titel:

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

Soft-Decision-Decodierung eines terminierten Faltungscodes

Soft decision decoding of a scheduled convolutional code

deren Beschreibung

(zutreffendes ankreuzen)
☐ hier beigefügt ist.
☒ am 01.12.1999 als
PCT internationale Anmeldung
PCT Anmeldungsnummer PCT/DE99/03824
eingereicht wurde und am ______
abgeändert wurde (falls tatsächlich abgeändert).

the specification of which

(check one)
☐ is attached hereto.
☒ was filed on 01.12.1999 as
PCT international application
PCT Application No. PCT/DE99/03824
and was amended on ______
(if applicable)

Ich bestätige hiermit, dass ich den Inhalt der obigen Patentanmeldung einschliesslich der Ansprüche durchgesehen und verstanden habe, die eventuell durch einen Zusatzantrag wie oben erwähnt abgeändert wurde.

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims as amended by any amendment referred to above.

Ich erkenne meine Pflicht zur Offenbarung irgendwelcher Informationen, die für die Prüfung der vorliegenden Anmeldung in Einklang mit Absatz 37, Bundesgesetzbuch, Paragraph 1.56(a) von Wichtigkeit sind, an.

I acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, Code of Federal Regulations, §1.56(a).

Ich beanspruche hiermit ausländische Prioritätsvorteile gemäss Abschnitt 35 der Zivilprozessordnung der Vereinigten Staaten, Paragraph 119 aller unten angegebenen Auslandsanmeldungen für ein Patent oder eine Erfindersurkunde, und habe auch alle Auslandsanmeldungen für ein Patent oder eine Erfindersurkunde nachstehend gekennzeichnet, die ein Anmeldedatum haben, das vor dem Anmeldedatum der Anmeldung liegt, für die Priorität beansprucht wird.

I hereby claim foreign priority benefits under Title 35, United States Code, §119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

German Language Declaration

Prior foreign appplications
Priorität beansprucht

			Priority Claimed	
19855453.2 (Number) (Nummer)	DE (Country) (Land)	01.12.1998 (Day Month Year Filed) (Tag Monat Jahr eingereicht)	☒ Yes Ja	☐ No Nein
(Number) (Nummer)	(Country) (Land)	(Day Month Year Filed) (Tag Monat Jahr eingereicht)	☐ Yes Ja	☐ No Nein
(Number) (Nummer)	(Country) (Land)	(Day Month Year Filed) (Tag Monat Jahr eingereicht)	☐ Yes Ja	☐ No Nein

Ich beanspruche hiermit gemäss Absatz 35 der Zivilprozessordnung der Vereinigten Staaten, Paragraph 120, den Vorzug aller unten aufgeführten Anmeldungen und falls der Gegenstand aus jedem Anspruch dieser Anmeldung nicht in einer früheren amerikanischen Patentanmeldung laut dem ersten Paragraphen des Absatzes 35 der Zivilprozeßordnung der Vereinigten Staaten, Paragraph 122 offenbart ist, erkenne ich gemäss Absatz 37, Bundesgesetzbuch, Paragraph 1.56(a) meine Pflicht zur Offenbarung von Informationen an, die zwischen dem Anmeldedatum der früheren Anmeldung und dem nationalen oder PCT internationalen Anmeldedatum dieser Anmeldung bekannt geworden sind.

I hereby claim the benefit under Title 35. United States Code. §120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, §122, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, §1.56(a) which occured between the filing date of the prior application and the national or PCT international filing date of this application.

PCT/DE99/03824 (Application Serial No.) (Anmeldeseriennummer)	01.12.1999 (Filing Date D, M, Y) (Anmeldedatum T, M, J)	(Status) (patentiert, anhangig, aufgegeben)	(Status) (patented, pending, abandoned)
(Application Serial No.) (Anmeldeseriennummer)	(Filing Date D,M,Y) (Anmeldedatum T, M; J)	(Status) (patentiert, anhängig, aufgeben)	(Status) (patented, pending, abandoned)

Ich erkläre hiermit, dass alle von mir in der vorliegenden Erklärung gemachten Angaben nach meinem besten Wissen und Gewissen der vollen Wahrheit entsprechen, und dass ich diese eidesstattliche Erklärung in Kenntnis dessen abgebe, dass wissentlich und vorsätzlich falsche Angaben gemäss Paragraph 1001, Absatz 18 der Zivilprozessordnung der Vereinigten Staaten von Amerika mit Geldstrafe belegt und/oder Gefängnis bestraft werden koennen, und dass derartig wissentlich und vorsätzlich falsche Angaben die Gültigkeit der vorliegenden Patentanmeldung oder eines darauf erteilten Patentes gefährden können.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true, and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

German Language Declaration

VERTRETUNGSVOLLMACHT: Als benannter Erfinder beauftrage ich hiermit den nachstehend benannten Patentanwalt (oder die nachstehend benannten Patentanwälte) und/oder Patent-Agenten mit der Verfolgung der vorliegenden Patentanmeldung sowie mit der Abwicklung aller damit verbundenen Geschäfte vor dem Patent- und Warenzeichenamt: *(Name und Registrationsnummer anführen)*	POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith. *(list name and registration number)* And I hereby appoint

Customer No. 26574

Telefongespräche bitte richten an: *(Name und Telefonnummer)*	Direct Telephone Calls to: *(name and telephone number)* Ext. ________
Postanschrift:	Send Correspondence to:

Schiff, Hardin & Waite
6600 Sears Tower 60606-6473 Chicago, Illinois
Telephone: +1 312 258 5780 and Facsimile +1 312 258 5921
or
Customer No. 26574

Voller Name des einzigen oder ursprünglichen Erfinders: Dr. THOMAS STURM 1-00	Full name of sole or first inventor: Dr. THOMAS STURM
Unterschrift ~~des~~ Erfinders Datum [signature] 21.3.2001	Inventor's signature Date
Wohnsitz MUENCHEN, DEUTSCHLAND	Residence MUENCHEN, GERMANY DEX
Staatsangehörigkeit DE	Citizenship DE
Postanschrift DAGLFINGER STRASSE 98	Post Office Addess DAGLFINGER STRASSE 98
81929 MUENCHEN	81929 MUENCHEN
Voller Name des zweiten Miterfinders (falls zutreffend):	Full name of second joint inventor, if any:
Unterschrift des Erfinders Datum	Second Inventor's signature Date
Wohnsitz ,	Residence ,
Staatsangehörigkeit	Citizenship
Postanschrift	Post Office Address

(Bitte entsprechende Informationen und Unterschriften im Falle von dritten und weiteren Miterfindern angeben).

(Supply similar information and signature for third and subsequent joint inventors).